University of Alberta

Digital Marbling Based on Computational Fluid Dynamics

by

Rüyam Acar ©

A thesis submitted to the Faculty of Graduate Studies and Research in
partial fulfillment of the requirements for the degree of Doctor of
Philosophy

Department of Computing Science

Edmonton, Alberta
SPRING 2005

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

0-494-08198-8

Your file Votre référence
ISBN:
Our file Notre reference
ISBN:

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

To Mom and Dad

Abstract

This thesis addresses the problem of using computational fluid dynamics techniques for artistic simulation purposes in computer graphics. Marbling is a traditional art of abstract painting which makes use of the dynamics of fluid flow. It involves complex physical and chemical processes, some of which have not even been investigated in conventional fluid dynamics. Inspired by the complexity and variety of traditional marbling patterns, and based on an analysis of the physical aspects of the marbling techniques, a generic tool has been developed that can be used to simulate a wide range of flows from the very viscous to the highly turbulent. Considering the numerical requirements for certain problems in fluid modeling, the digital marbling problem is first divided into two parts: the fluid model and the transport model. The fluid model involves solving the fluid equations; it consists of two parts. The first part constitutes a multiscale model based on mesoscale dynamics and viscous fluid equations; it allows for various user control to simulate complex flow behaviours as in traditional marbling techniques as well as laminar and turbulent flows. The second part is based on a solution of inviscid equations; it is used to simulate vortex sheet dynamics, turbulence, and other artistic effects (including some well-known marbling techniques). The transport model describes the material transport throughout the velocity space. It includes an interface model and two reaction models to simulate certain characteristics of marbling flows as well as other artistic effects.

Acknowledgments

I would like to thank my supervisor, Pierre Boulanger, for his support and patience. I greatly appreciate his encouragement and confidence in my ability. His suggestions have much improved the presentation of this thesis as well as my writing skills. I would also like to thank the members of my committee, John Bowman, Jason Carey, Bruno Eckhardt and Herb Yang for showing interest in my work, and for valuable feedback on this thesis. I acknowledge the support of my former supervisors, John Buchanan and Mark Green, in the beginning of this project. I also wish to add my thanks to Martin Jagersand for his time and effort in reviewing some of the earlier documentation of this research. Thanks also to all my friends for being there for me when I needed them and for making my years away from home bearable.

I could not have completed this thesis without the support of my family. First, I would like to thank my sisters for the inspirational phone conversations and all the e-mails we exchanged over the years; they helped me immensely to get through my years in graduate study. My gratitude to my parents cannot be expressed in a few words. They have always been a great source of inspiration for me. Their example and unconditional support have provided the enthusiasm and the driving force in every goal I pursue.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Computer Art 2

1.3 Problem Definition and Methodology 3

  1.3.1 Multiscale Fluid Model 5

    1.3.1.1 Discretization of Fluid Equations Including Mesoscale Fluctuations. 5

    1.3.1.2 Simulation of Fluctuating/Turbulent Motion 7

  1.3.2 Vortex Sheet Model 7

  1.3.3 Transport Model 8

    1.3.3.1 Advection Model 8

    1.3.3.2 Interface Dynamics 9

    1.3.3.3 Density Fluctuations 10

1.4 Thesis Contributions 10

1.5 Overview 13

2 Traditional Marbling **14**

2.1 The Basic Technique 14

2.2 Materials 14

2.3 Physical Processes 15

2.4 Patterns 17

3 Fluid Modeling at the Mesoscopic Level **20**

3.1 Microscopic Level 20

3.2 Macroscopic Level 21

3.3 Mesoscopic Level 22

  3.3.1 Lattice Based Models 22

  3.3.2 Stochastic Models 23

    3.3.2.1 Dissipative Particle Dynamics 24

4 Discrete Fluid Model **28**

4.1 Spatial Discretization Techniques for Hydrodynamic Equations 29

  4.1.1 Finite Difference Method 29

  4.1.2 Finite Element Method 29

4.1.3 Finite Volume Method 30
4.1.4 Gridless Methods 30
4.1.5 Spectral Methods 32
4.2 Discretization of Navier–Stokes Equations In Computer Graphics: A General Overview 33
4.3 Voronoi Fluid Model 34
4.3.1 Momentum Equation With Mesoscale Fluctuations 34
4.3.2 Voronoi Grid Structure 36
4.3.3 Spatial Discretization of the Momentum Equation 39
4.3.4 Temporal Discretization and Stability Issues 43
4.3.5 Incompressibility 44
4.3.5.1 Artificial Compressibility Methods 45
4.3.5.2 Fractional Step Methods 45
4.3.5.3 Voronoi Based Finite Volume Projection 47
4.3.6 Boundary Conditions 49
4.4 Interpolation and Smoothing of Fluid Vectors 52
4.4.1 Scattered Data Interpolation Techniques 52
4.4.1.1 Moving Least Squares 52
4.4.1.2 Splines (Radial Basis Functions) 53
4.4.1.3 Kriging 55
4.4.2 Kriging in the Marbling Model 57
4.4.3 Summary 61
4.5 Transport Model 61
4.5.1 Finite Difference Methods for the Advection Equation 61
4.5.2 Semi-Lagrangian Advection 64
4.5.2.1 Spatial Interpolation 65
4.5.2.1.1 Experimental Results 67
4.5.2.2 Sharp Interface Transformation 68
4.5.3 Reaction 72
4.6 Interface Dynamics 73
4.6.1 Numerical Modeling 74
4.6.1.1 Front Tracking (Free Boundary) Methods. 74
4.6.1.2 Front Capturing Methods 75
4.6.2 Diffuse-Interface Model 77
4.6.2.1 Cahn-Hilliard Theory 77
4.6.2.2 Numerical Solution of the Cahn-Hilliard-Advection Equation 81
4.6.2.3 Cahn-Hilliard/Navier–Stokes Coupling 82
4.6.2.4 Nonlocal Effects 83

5 Digital Marbling 87
5.1 Laminar Mixing 88
5.1.1 Mechanical Mixing (Stirring) 88
5.1.1.1 Characterizing Chaos in 2D 88
5.1.1.2 Experiments 90

5.1.2 Interfacial Events 91
  5.1.2.1 Experimental Results 91
5.2 High Reynolds, Turbulent and Fluctuating Motions in Digital Marbling 94
5.2.1 What is Turbulence ? 97
5.2.2 Vortex Sheet Simulation 98
  5.2.2.1 Numerical Modeling 98
  5.2.2.2 Level Set Vortex Sheet Model 102
    5.2.2.2.1 Discretization 103
    5.2.2.2.2 User Control 104
    5.2.2.2.3 Experimental Results 105
5.2.3 Multiscale Simulation of Turbulence and Fluctuating Motion Based on a Mesoscale Model 125
  5.2.3.1 Statistical Properties of Turbulence 129
  5.2.3.2 Conventional Turbulence Modeling 130
  5.2.3.3 Turbulence Modeling in Computer Graphics 131
  5.2.3.4 Multiscale Fluid Model for Marbling 131
    5.2.3.4.1 User Control 134
    5.2.3.4.2 Some Implementation Issues and Discussion 136
    5.2.3.4.3 User Interface 137
    5.2.3.4.4 Experimental Results 137
5.2.4 Density Fluctuations, Turbulent (Eddy) Diffusion. 145
  5.2.4.1 Density Fluctuations in Digital Marbling 150
    5.2.4.1.1 Experimental Results 150

6 Conclusion **155**
6.1 Summary 155
6.2 Future Work 157
6.2.1 Experiments 157
6.2.2 Further improvements 158
6.2.3 Extensions 158
6.2.4 Future Research Directions 159

List of Symbols

$\mathbf{P}$	Momentum
η	Viscosity
ρ	Density
ν	$\frac{\eta}{\rho}$, kinematic viscosity
σ	Noise (random force) amplitude
γ	Friction constant (amplitude of dissipative forces) in stochastic Langevin (DPD) equations
ξ	Gaussian stochastic variable
dW	Wiener process increment
$\mathcal{L}$	$\sim \frac{\partial}{\partial \mathbf{P}}$, Lioville operator
$\mathbf{n}_{ij}$	$\frac{\mathbf{r}_{ij}}{r_{ij}}$, unit vector between pointing from particle i to j or Voronoi face normal between cells i to j
w	Local weight function
r_c	Cut-off range in localized weight functions
k_{th}	Thermal conductivity
T_{ij}	Temperature difference between cells (particles) i and j
$\bar{T}_{ij}$	Average temperature between cells i and j
$\mathbf{\Psi}$	Basis functions in corrective SPH
ε	Least square error functional
ς	Strength of density oscillations in SPH equation of state
ζ	Artificial compressibility parameter

ϕ	Projection scalar (pressure correction term)
β	Pressure gradient influence in projection method
$\bar{\epsilon}$	Convergence parameter in Voronoi projection method
Ω	Domain boundary
λ	Radial basis function (RBF)/kriging interpolation coefficients
ι	Smoothing parameter in RBF splines
$E\{X\}$	Mean value of X
σ_e^2	Error variance (least squares minimization value) in kriging
ℓ	Lagrange multiplier in ordinary kriging
$\hat{\sigma}_e^2$	Modified error variance in ordinary kriging including the Lagrange multiplier effect
$\hat{\sigma}_\eta^2$	Nugget value in kriging
$\hat{\epsilon}$	Constant used to prevent division by zero errors
χ	Fractional value used in bilinear interpolation
$\dot{\upsilon}$	Shear rate
σ_s	Surface tension coefficient.
Ca	Capillary number.
C_ϕ	Interface parameter in front capturing methods.
C	Advected color/concentration/ phase parameter.
κ	Interface curvature.
$\delta()$	Delta function.
$f_\varphi(C)$	Free energy function
μ	Interface potential energy
$\hat{\xi}$	Interface thickness
M	Interface mobility
$\hat{\mu}$	Interface potential
χ_s	Fractional splitting parameter in spectral solution of Cahn-Hilliard equation

$\dot{C}$	Concentration average in a local area
$\dot{\alpha}$	Amplification factor that determines the effect of *nonlocal effects* in diffuse-interface equations
ω	Vorticity
ψ	Stream function
Γ	Vortex sheet strength
φ	Vortex sheet level set parameter
δ_ε	Vortex sheet width
ϵ	Energy dissipation rate
$E(k)$	Energy spectrum
μ_e	Eddy viscosity
θ	Velocity limit that defines random force application in multiscale fluid model
α	Amplification of the magnitude of the fluctuations
D	Diffusion coefficient
D_ε	Eddy diffusivity

Chapter 1

Introduction

1.1 Motivation

Visual arts have always been influenced by the developments in science and technology. Most of the great artists were involved in the scientific aspects of their artistic environment; they studied human visual perception, physical properties of different media and explored new tools. Renaissance artist Leonardo da Vinci is well-known for his studies in human anatomy and fluids. In the 19th century, impressionists Monet, Van Gogh, Seurat studied color science and physics of light. 20th. century artist Escher's works demonstrate the integration of mathematics and art. In the last few decades computers have been accepted as a new aesthetic medium and begun changing the concept of painting. Computers differ from the earlier media in several aspects. It is a flexible environment; it can simulate any other medium without suffering from most of the difficulties that occur in its real world counterpart. In addition, it opens the world of mathematics and algorithmics to the service of artists extending the limits of representation medium and creativity. With the computational power of computers, digital paint systems can provide high-level, creative and complex tools for the artists which would not be available in traditional media.

Marbling is a traditional art of paper decorating and an intuitive painting style for creating abstract designs, and figurative patterns. The technique is based on forming patterns on the surface of a liquid by controlling the flow of pigments that are dripped or sprinkled onto the surface. The beauty of the swirling, wavy patterns lies in the unpredictable, dynamic nature of fluid flow, which also makes every design unique. By making use of dynamics of fluid flow and controlling the flow properties by adding certain substances to the liquids, a large variety of patterns can be created. The marbling process involves complex fluid behaviours, both in terms of fluid dynamics and intermolecular physics. It is also a very delicate process where designers have to deal with many difficulties caused by the chemical reactions in the materials and by the dynamic nature of the fluid flow which cannot be easily predicted or repeated.

Computer marbling is advantageous mainly in two aspects. First it provides a more flexible environment in which most of the difficulties that occur in traditional marbling can be removed. For example the user can change the flow properties during the animation or reverse the operation.

Especially when the viscosity is low, fluids move very fast and the flow becomes very difficult to control. In real marbling the user has to choose one moment to stop the process. Most of the details on the surface are lost or the patterns are distorted during the transfer to the paper. In digital marbling, the user has a control over the whole process. In addition he can design boundary configurations (as combs) which are difficult to build in real life. Another advantage of the computer version is that the user can model interesting fluid behaviours which are not easy to produce in real life, such as highly turbulent flows. Moreover the user can create flows with high complexity and even flows that are not necessarily physically realistic.

Computational fluid dynamics has been an active research area for more than a century, but several flow problems still cannot be modeled with the standard equations. Considering the complexities in the marbling flows and the difficulties in numerical fluid modeling, it may not be possible to simulate all the details in marbling with complete accuracy with the current knowledge and computer technology. However, taking advantage of computational methods, a model that provides similarly complex fluid behaviors can be developed.

The main objective of this thesis is to provide a new tool for artists not only by directly simulating the real marbling process, but also, based on the same principle, by making use of certain features of fluid dynamics and mathematical methods.

The studies related to this work can be analyzed under two different areas: fluid modeling and computer art. Technically, fluid modeling studies constitute the primary relevance to this thesis. Fluid modeling research in computer graphics mostly aims at simulating realistic environments. Most fluid studies in computer graphics are applications of existing Computational Fluid Dynamics (CFD) [50, 141, 140] and Computational Physics methods, very few present novel approaches for CFD and graphics [53]. Both computer graphics and CFD studies will be discussed in more detail throughout the document; in this chapter only a general perspective will be presented based on their relation to this work. The second category, is related to the digital marbling problem only in the broad context of generating artistic images by using computational capabilities of computers. This is usually accomplished either by providing creative tools for artists or by the automation of some artistic techniques. Since it was some of these research that first inspired me to pick this thesis topic, I would first like to present a brief review in this area.

1.2 Computer Art

Using mathematical algorithms is not new for computer artists; several artists such as Ken Musgrave [76], Yoichiro Kawaguchi [77] have been using fractals and procedural textures to create images (see SIGGRAPH electronic arts archives [88] for several others). Computer graphics work in this context is mostly based on imitating or simulating the real media. One of the earliest of those work was Strassman's Sumie painting model [145]. Strassman modeled the paintbrushes in the traditional Japanese art of Sumie by simulating the physical characteristics of the brush bristles and the ink. He designed a painting and an automatic rendering system with this technique. Later Small [136] simulated the movement of water and pigments in watercolors through the use of cellular automata. Curtis et al. [29] extended this work to create a more realistic and sophisticated watercolor and paper model. They also provided a watercolor rendering of 3D objects.

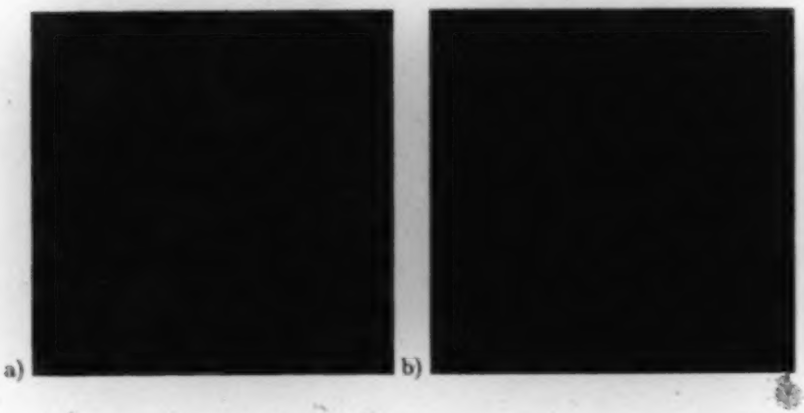


Figure 1.1: Comb and swirl patterns created with Fractal Design Painter.

Other research works [69, 135, 111] in non-photorealistic rendering are also related to the simulation of painting techniques. Non-photorealistic rendering refers to algorithms that generate creative and expressive images rather than producing photographic images. Computer graphics research have mostly focused on producing high quality photorealistic images. However, photorealism is not necessarily the ideal type of representation. Artists use abstractions to express aesthetic impressions. There are also some applications in technical illustration in which photographic images provide too much information and do not convey the general visual information. For this reason, various media such as pen and ink, watercolors, and charcoal are used to create interpretive images. Haeberli's [69] painterly rendering system can be considered as the pioneering work in non-photorealistic rendering. He presented an interactive image processing technique and a rendering method to create abstract images. Basically, he sampled and rendered images with brush strokes to create renderings in an impressionist style. Meier [111] combined this image processing idea with particle rendering to develop a technique for a painterly rendering of animations. Sims [135] demonstrated a good example for generating aesthetic, abstract images with procedural methods. He applied the evolution techniques of variation and selection on symbolic expressions to create complex, abstract textures, and motions.

Some commercial painting programs such as Fractal Design Painter [54] provide marbling effects. They simulate very viscous marbling types where the boundary influence is local (Figure 1.1a shows a comb pattern created by Fractal Desing Painter). Fractal Design Painter also provides some liquid effects such as a swirling motion around a cursor (Figure 1.1b). These effects do not rely on physics based fluid modeling and are mostly generated by automatic functions. Although the focal point of this thesis will be mainly on realistic and dynamic fluid motions whose complex and unpredictable nature can only be described using fluid dynamics, these comb patterns can also be simulated with the proposed technique.

1.3 Problem Definition and Methodology

In traditional marbling, various substances are used to adjust the viscosity, surface tension and rheological properties of the fluids. From a physical modeling perspective, these parameters define a wide range of flow types from very dynamic to very viscous flows.

An important characteristic of the marbling flow is that it exploits the surface tension properties of the pigmented solutions to make the paints stick to the surface. As a result, **marbling flow can be considered two-dimensional**. This leaves out the modeling of 3D effects such as waves and free surface flows. Thus, in surface modeling, only the motion of liquid-liquid interfaces will be considered. Furthermore, marbling paints are usually immiscible, or slightly miscible, that is, they do not diffuse or mix with each other; sharp interface lines between the paints is a characteristic of most marbling designs. Accordingly, transport model should be in control of the diffusive property of the interface.

Background and pigment solutions constitute colloidal and polymeric suspensions. The properties of these fluids are determined by the events taking place at mesoscopic levels. For some of these events and their effect on the continuum level there is no deterministic descriptions. For this reason, stochastic models have been popular for the simulation of these complex fluids; in statistical modeling the complex motion at the mesoscopic level is described by stochastic processes incorporating ideas from Brownian dynamics. On the other hand, some aspects of the colloidal and polymeric liquids are well defined; considering the traditional marbling patterns which makes use of the interfacial lines between different fluids, some of these properties can be described by interface free energy dynamics based on van der Waals theory.

Given these, digital marbling problem will be studied in terms of the properties of the fluid and transport equations. In order to simulate fluid complexity at different levels as well as turbulent motion, fluid equations including mesoscopic fluctuations will be used. Without these fluctuations, fluid equations can only describe basic viscous flow properties. Futhermore, at high Reynolds numbers, vortex sheet modeling will be used. Vortex sheet simulation involves the flow patterns created by boundaries in high Reynolds number (but not necessarily turbulent) flows. In transport model, advection properties and the effect of interfacial events will be covered. Overall, digital marbling model will cover a wide range of flows, not only limited to the traditional marbling examples.

A physical description of the flows in each category together with the numerical requirements will be given in the following section, but first an outline of the algorithm will be presented.

General Structure of the Proposed Algorithm

The fluid equations are discretized on a relatively coarse Voronoi grid, and the transport equations are discretized on the underlying higher resolution rectangular grid (which will also be referred to as the image grid). The connection between the fluid and image grids is obtained by a kriging interpolation technique. In other words, the velocity vectors on the fine grid are obtained by interpolation. Kriging is an irregular data interpolation method that provides interpolation and smoothing of the data with the same set of equations. It provides the highest interpolation quality in terms of smoothness, continuity and accuracy. Basically, the incompressible viscous fluid solution can be considered as a hydrodynamic description for the background liquid in marbling. Other properties including turbulent and fluctuating motions are built on this numerical background. Vortex sheet simulation is based on a special case of inviscid equations which are discretized independently from the Voronoi fluid model, on a rectangular grid with the same resolution as the image grid. Figure 1.2 shows a block diagram of the model.

1.3.1 Multiscale Fluid Model

In this thesis, I propose a multiscale statistical model based on mesoscale dynamics to simulate a wide range of complex flow behaviours from smooth oscillations to highly turbulent flows.

1.3.1.1 Discretization of Fluid Equations Including Mesoscale Fluctuations.

The incompressible fluid equations are discretized on the Voronoi grid using Finite Volumes. Fluctuations are incorporated into the momentum equation based on mesoscale modeling [131, 46]. A Voronoi grid is chosen to allow one to adopt the physical properties of the original mesoscale models on which the marbling fluid model is based. This mesoscopic model is obtained by discretizing the microscopic or macroscopic fluid equations on a Voronoi structure as it provides an efficient space allocation and more importantly for the numerical simplifications it allows in derivation due to its geometric properties. Thus, in order to preserve the hydrodynamic balance of the mesoscopic forces, I also maintained the Voronoi structure in the fluid model. Moreover the Voronoi grid structure, in its mainly hexagonal form, is advantageous for its symmetry properties which becomes more effective in turbulence and vortex simulations. It is also beneficial for its adaptibility to complex geometries.

In viscous flows at large scales, the fluctuations disappear and the momentum equation reduces to the Navier–Stokes momentum equation. Discretization of the Navier–Stokes equations on regular grids (for the simulation of laminar flows) has been widely used in computer graphics. The use of the Navier–Stokes equations based on explicit finite difference schemes [18, 50, 51] usually suffers from instabilities. Later, Stam [140] introduced a stable solution based on semi-Lagrangian advection and semi-implicit time integration. However on a Voronoi structure, finite volume discretization is a more appropriate solution both in terms of accuracy and implementation efficiency. In this thesis, a fourth-order Runge-Kutta scheme was used for time integration and a second order projection method was formulated on the Voronoi structure for the solution of the continuity equation in order to enforce incompressiblity. To further improve the stability conditions a kriging smoothing is applied. Although still conditionally stable, this level of flexibility was sufficient in the current model. To achieve unconditional stability a fully implicit finite volume discretization or a semi-implicit fractional step method could be considered. However in either case, the implicit matrix formulation based on the Voronoi grid would be computationally expensive and complex.

At this point, I would like to emphasize that the mentioned Navier–Stokes solutions are only suitable for simulating viscous flows and modeling high Reynolds number flows requires a more accurate discretization at high resolutions. Turbulent flows are very sensitive to small fluctuations; therefore, discretization of these flows can not tolerate numerical diffusion as it would damp out the high frequency information.

Finally, all the boundaries including the borders and the external moving objects representing the marbling combs are defined as immersed boundaries with no-slip conditions. Boundaries with

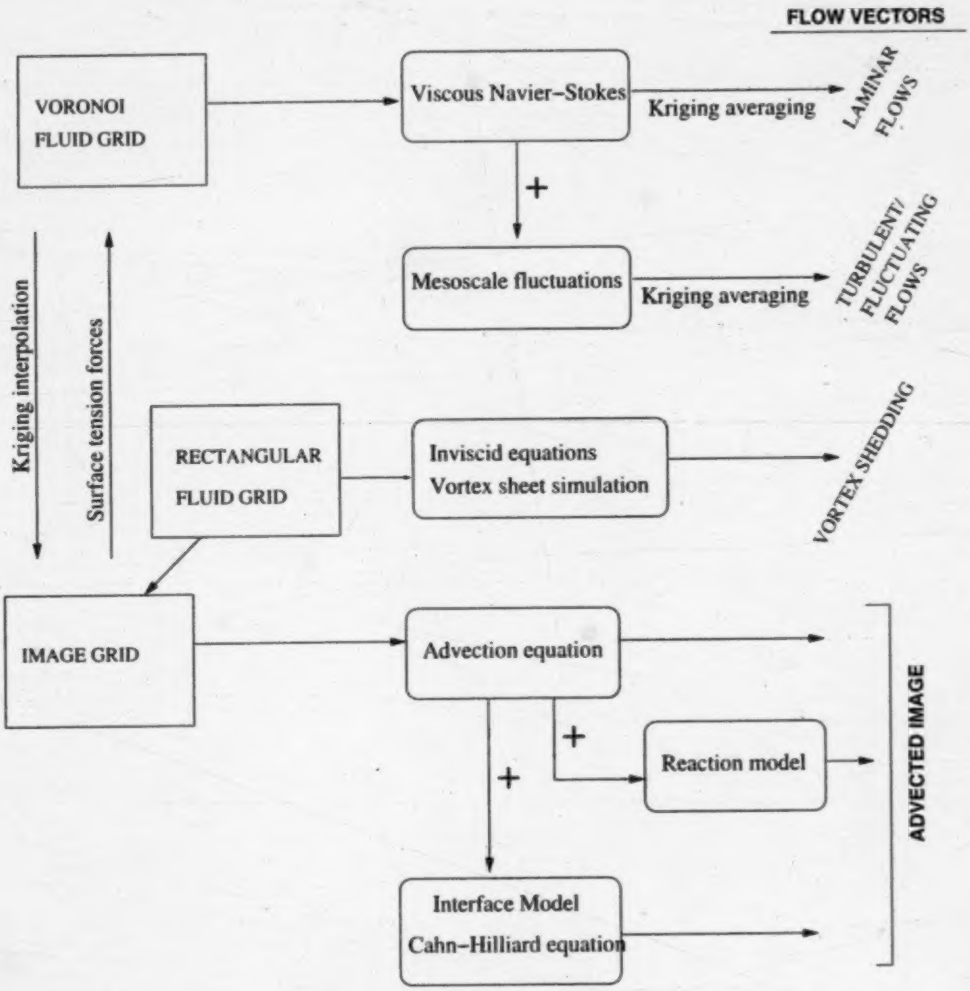

FLOW VECTORS
VORONOI FLUID GRID
Viscous Navier–Stokes
Kriging averaging
LAMINAR FLOWS
+
Mesoscale fluctuations
Kriging averaging
TURBULENT/ FLUCTUATING FLOWS
Kriging interpolation
Surface tension forces
RECTANGULAR FLUID GRID
Inviscid equations
Vortex sheet simulation
VORTEX SHEDDING
IMAGE GRID
Advection equation
+
Reaction model
+
Interface Model
Cahn–Hilliard equation
ADVECTED IMAGE

Figure 1.2: Digital marbling model.

multiple points and different geometric shapes can be designed by the user.

1.3.1.2 Simulation of Fluctuating/Turbulent Motion

In order to model the effect of fluctuations on the mean flow, fluid equations including the mesoscale fluctuations are averaged by kriging. In traditional turbulence modeling, which is mostly based on statistical approaches, the influence of small scale fluctuations on the mean flow is formulated through additional equations [16] obtained by averaging the fluid equations that consider the effect of a random component in the flow variables. In the marbling model, fluctuations are included in the solution directly. In this way, the user will have more control on the simulation to create various effects. The range and the homogeneous and isotropic characteristics (that is, Brownian nature) of the fluctuations in the original mesoscale model have been modified. They can be conditional: the user can define the way in which the fluctuations are triggered. Application of the momentum forces with the random fluctuations and filtering are repeated recursively. By adjusting the level of these recursions and the compressibility level, together with the flow properties such as the ratio of the random and dissipation coefficients, the user can control the effect of fluctuations on the mean flow at different levels. For example the fluctuations can cause smàll oscillations or more turbulent motion. In this way, most of the unpredictable, dynamic flows in traditional marbling as well as highly turbulent flows can be simulated with the same model. In computer graphics, there have been some studies addressing the turbulent motion mostly focusing on gaseous phenomena. Random forces in Navier–Stokes has been considered in Stam and Fiume's work [138] to model gaseous phenomena; here the random fluctuations don't have a physical connection to the mean flow. In some methods, temperature [52] or density driven [42] forces have been included in the momentum equation (some in accordance with the conventional turbulence modeling). Or in other works, vorticity confinement is used to compensate for the diffusive behaviour of the numerical solutions and consequently to maintain vorticity properties, again mostly to simulate gasses and smoke. A generic turbulence model based on statistical averaging has not been introduced in computer graphics. Furthermore, although recent mesoscale methods consititute a potential for turbulence modeling they haven't been used for this purpose, in computational physics studies to my knowledge.

1.3.2 Vortex Sheet Model

Vortex sheet simulation involves a more realistic simulation of the boundary influence in high Reynolds number flows. When Reynolds number is within a certain range, above the laminar state and below a level where the flow becomes turbulent, the boundary perturbation depending on its surface geometry creates organized instability patterns called vortex streets. Direct numerical simulation of these flows with the Navier–Stokes equations is very difficult; this problem is usually solved in CFD [94] with high-order methods, by refining the grid around the boundaries and vortex regions. This approach is computationally very expensive especially if the boundary is moving. A more common approach is to evolve the vorticity in the inviscid limit. These can generally be categorized as Eulerian and Lagrangian methods. Vortex sheet problem can be considered a special case of inviscid equations where the vorticity is concentrated in a region. The main difficulty is to model the singularity (to define the curl of the velocity) numerically at the vortex sheet surface. In the

Eulerian approach, it is difficult to capture the high-resolution evolution of the vortex sheet due to the fixed grid representation and numerical diffusion. In the Lagrangian vortex methods, particles can easily adapt to the vortex sheet structure; however, it is difficult to keep the particles organized in highly detailed vortex structures. Besides, representing continuous velocity fields with discrete particles is always problematic. In this thesis, an Eulerian level set based solution is proposed to the vortex sheet problem, following the elegant approach introduced in [70] in CFD. The main idea of this method is to represent the vortex sheet surface by a smoothed level set function. Smoothing functions together with numerical diffusion are used to stabilize the vortex sheet singularity. With an appropriate numerical discretization where a semi-Lagrangian method is used for the advection of the level set and a Fast Fourier Transform spectral method is used for the Poisson equation, this novel method becomes very practical and efficient both in terms of computation time and implementation. It provides a flexible environment that allows the user to control several fluid properties and create interesting effects. This method is used to model realistic vortex sheet evolution, two dimensional vorticity dynamics and turbulence (including the interaction of vortices and vortex sheets) and finally viscous comb patterns.

1.3.3 Transport Model

Material transport in a flow is described by the advection-diffusion equation. Diffusion is usually neglected in marbling experiments due to the immiscible characteristics of most marbling paints. Also two reaction models are introduced, one is used to model material expansion and the other to simulate density fluctuations.

1.3.3.1 Advection Model

For the solution of advection equation, again the semi-Lagrangian method has been the most popular choice in computer graphics for its unconditional stability. One disadvantage of this approach is that it produces numerical diffusion as it relies on the interpolation of the related data. In their first model [140], Stam used linear interpolation in semi-Lagrangian advection which was highly diffusive. Later Fedkiw et.al. improved this method in [43] by using a Hermite cubic interpolation technique. Although cubic interpolation still caused diffusion, it was considered acceptable in simulating gaseous phenomena.

In this thesis, also the semi-Lagrangian approach has been followed for the advection solution. In order to minimize the numerical diffusion and increase the interpolation accuracy, 5th order B-splines were used. B-splines provide the optimum solution in image reconstruction since the B-spline filter approximates the ideal sinc filter, by minimizing the interpolation error, as its degree is increased. However B-splines are not free from oscillations; in my experiments it was observed that the oscillatory behaviour became prominent especially after the 5th order. Another disadvantage of B-splines is that they can be computationally expensive as the order increases, however a fast algorithm as proposed by [153, 154, 155] was used instead. This method improved the results both in terms of diffusion and level of detail compared to the Hermite cubic interpolation proposed by Fedkiw et.al. [43] in computer graphics. However unless an ideal sinc filter is used, diffusion is inevitable, so the B-Spline interpolation does not completely prevent the blurring that occurs in material deformation, especially in high-frequency regions and at high resolutions. In order to be able to simulate very sharp interface motion, even jagged edges in some marbling patterns, an alternative solution based on a tangential transformation idea (as similarly used in CFD by Yabe et al. [165]) is proposed. In this method advection is performed on the transformed values of the concentration value. The real values are obtained from the inverse transformation, which is designed to control the sharpness of the interface. This method is simple and practical, compared to the level-set-based method which has been recently proposed in computer graphics [160] to maintain the sharpness of the advected image.

1.3.3.2 Interface Dynamics

Interface model includes the surface tension effects and topological transitions such as breakup and coalescence. In the marbling model, a diffuse-interface method is used for modeling the interface dynamics. The diffuse-interface method, as in other front capturing models, such as the level-set method [147] and volume of fluid method (VOF) [72], describes the interface between two fluids with a phase parameter. However, here the phase parameter is based on the interface physics.

Evolution of the phase parameter is described by the Cahn-Hilliard [15] theory based on the diffusion of an interface potential which is derived from the calculus of variations to minimize the interface free energy according to van der Waals theory. This potential value consists of a free energy and a gradient term which balance each other as the interface evolves. This Cahn-Hilliard term is included in the advection equation. Operator splitting was used to solve the advection and diffusion (Cahn-Hilliard) terms separately. Cahn-Hilliard equation is numerically challenging due to its stiff characteristics. An unconditionally stable linearized semi-implicit spectral method, as proposed in [40] in CFD, was used to solve this problem.

Unlike other interface methods [72, 147], the parameters of this system can be adjusted to change the mobility, diffusivity and sharpness of the interface. Interface equation is coupled with the Navier–Stokes equations to model surface tension forces. This is accomplished by the inclusion of an additional surface tension term in the momentum equation that depends on the interface potential and the gradient of the phase parameter.

The diffuse-interface implementation is simple and more practical compared to the reinitialization process applied in the level set method [148] or to the marching cubes algorithm used in VOF approach [73]. In diffuse-interface method the interface evolves smoothly and continuously, and topological transitions are handled naturally. This system is also flexible in the choice of the free energy which can be defined according to the rheological properties of the fluid. One disadvantage of this method is that the evolution of diffusion equation is very slow. In order to be able to speed up this process, and to increase the elastic energy in the system, a method based on modifying the interface energy according to the local density average was developed. The phase-separation property of the Cahn-Hilliard equation has been used for vector field visualization in computer graphics [63]; however the diffuse-interface approach for modeling fluid interfaces has not been introduced for computer graphics to my knowledge.

1.3.3.3 Density Fluctuations

This part involves the simulation of density fluctuations and turbulent diffusion, which probably have a considerable effect on most marbling flows. A practical but effective approach based on the multiscale fluid model to simulate the oscillations at different scales is proposed. In this method a reaction term which directly depends on the velocity fluctuations is included in the advection term. This term is composed of a sinusoidal function which depends on the previous positions in the Lagrangian trajectory of the particles. Backtracked positions which are defined for the semi-Lagrangian advection discretization are also used for this purpose. A sinusoidal function is used to obtain a smoothed variation. As a result, oscillations in the velocity are reflected in the density. The idea that the density fluctuations are induced by mesoscale eddies complies with the description of turbulent diffusion. However conventional modeling of turbulent diffusion is mostly predicated on closure models which aim to describe the properties of a diffusion coefficient. In computer graphics, a temperature-based reaction modeling approach as used in [139] for the simulation of fire can be considered relevant. With the current method, one can easily simulate various effects from smooth density oscillations to highly turbulent fluctuations (as in turbulent diffusion). These can be used to simulate similar oscillations as in some traditional marbling flows or turbulent mixing of dissimilar fluids or even other reacting flow effects such as combustion.

1.4 Thesis Contributions

The main contributions of this thesis are:

Concepts

- *The realization of a "digital marbling" simulation system using fluid dynamics as an artistic tool for computer graphics.* Some marbling styles, which do not require the solution of fluid equations, such as typical comb patterns have been simulated in painting tools with automatic functions. In addition, laminar flow simulation based on fluid equations and some aspects of turbulence modeling, though mostly with application specific approaches, have been considered in computer graphics. However marbling involves a wide range of fluid behaviours and complex physical processes, some of which have not even been described in theory. I have analyzed traditional marbling according to its physical properties in terms of fluid modeling and developed a generic tool based on several aspects of fluid dynamics, including the flows related to traditional marbling as well as others with interesting properties.

- *Introducing mesoscopic fluid modeling into computer graphics.* Considering the colloidal properties of marbling fluids, I used mesoscopic fluctuations to obtain complex, detailed fluid behaviours. Although, there are also deterministic mesoscale models, the stochastic approach is chosen as a basis for the model to take advantage of the flexibility of the random description in creating different effects.

Development of a Multiscale Fluid Model

- *The adaptation of a mesoscale fluid model for the use of computer graphics simulation.* Among the stochastic mesoscale methods, a model with higher accuracy and that is suitable for scaling in terms of its physical properties was selected. This model is based on a discretization on a Voronoi structure, which also makes it suitable for adaptation to the Eulerian frame. Stochastic mesoscopic models are mostly based on a Lagrangian description; this is mainly to be able to model certain properties of complex fluids at small scales. However since details of the fluid behaviours in these scales are not relevant to marbling simulation and as adaptive regridding of the Voronoi structure is a very expensive task in any case it was decided to use the Eulerian description. I have analyzed the relations between the discretization of these mesoscale models and the gridless and finite volume discretization methods for the macroscopic fluid equations. Based on this analysis and also considering the requirements of the marbling flows I made the following general modifications to the original mesoscale fluid model. Adaptation to the Eulerian frame. Enforcement of the continuity condition by changing the pressure description (as it was based on a compressible description in the original models).

- *Finite volume discretization of hydrodynamic equations including mesoscale fluctuations based on a Voronoi structure.* While discretization of fluid equations on regular grids has been widely covered in computer graphics, finite volume formulation on a Voronoi structure have not been considered. Voronoi structure can be advantageous in a graphics application mainly for its adaptability to complex surfaces. Although Voronoi structure is mainly chosen to achieve compatibility with the mesoscale discretization, with a mainly regular hexagonal structure, it is also advantageous for its symmetry. But more importantly for graphics modeling it is advantageous for its adaptability to complex surfaces. I have developed the optimum discretization on the Voronoi structure, both for the momentum and continuity equations, in terms of computation and and implementation efficiency. Stability limit is improved with 4th-order Runge-Kutta time integration and kriging smoothing.

- *Fluid modeling based on multiscale averaging of mesoscale fluctuations.* In order to average the mesoscale fluctuations, I have developed a kriging smoothing method based on the Voronoi structure. I have developed an algorithm which defines the multiscale properties of the fluid in recursive repetition of the momentum, divergence, and averaging steps. I modified the originally Brownian characteristics of the mesoscale fluctuations by changing their range and also by making them conditional. Together with the properties of the random and viscous forces this algorithm allows the user to control incompressibility and viscous properties of the fluid as well as the range, scale, smoothness and condition of the oscillations. This level of user control at every scale of the flow evolution is a significant property of the model developed in this thesis especially from a graphics modeling perspective. This model allows the simulation of a wide range of complex fluid behaviours, from smooth oscillatory motions to highly turbulent flows, which cannot be modeled with the fluid models proposed in computer graphics. A similar turbulence approach based on mesoscale dynamics has not been proposed in CFD either.

- *Development of a user interface for the Voronoi fluid model.* I have developed a user interface where only the basic parameters of the multiscale fluid model can be adjusted.

Transport Model

- *Advection equation:* I improved the semi-Lagrangian advection method (which is commonly used in graphics applications) by applying 5th order B-splines in the spatial interpolation. This decreases the numerical diffusion considerably while improving the level of details. In order to be able to control the interface sharpness completely I introduced a method based on the tangential transformation of the interface parameter.

- *Reaction model:* I have introduced two reaction models. The first one is a bistable system which models the expansion of material in the flow direction. The other one is for modeling concentration fluctuations and eddy diffusion. It is built on the semi-Lagrangian advection solution to reflect the velocity fluctuations on to the concentration evolution. This model can be used to add details of the material advection by modeling smooth oscillations or it can be used to simulate highly turbulent density fluctuations.

Interface Dynamics

- In order to simulate fluid interface properties I have introduced a diffuse-interface method. Unlike the previous approaches proposed in graphics, this model is based on interface physics. It describes the interface evolution with a high order diffusion equation, (Cahn-Hilliard equation), based on the interface free energy. Thus, it allows the modeling of interface events such as breakup and coalescence naturally. I have also implemented a surface tension force on the Voronoi fluid grid to model the coupling of Cahn-Hilliard and Navier–Stokes equations. In order to speed up the slow evolution of diffusion dynamics, I have developed a method based on modifying the interface free energy according to the local density average.

Vortex Sheet Simulation

- Fluid boundary interactions at high Reynolds numbers produce very interesting vortex sheet patterns. Modeling of vorticity equations have been considered in computer graphics, however they either suffer from numerical diffusion or the discontinuities in particle representation. Vortex sheet is a special case of vorticity dynamics which is difficult to model due to its surface singularities. In order to model the vortex sheet evolution I have introduced a model specifically based on vortex sheet dynamics. This is a very practical and simple solution based on a smoothed level set idea and a fast, stable discretization which makes it suitable for graphics modeling. I have also suggested various uses of the level set parameters to create different effects other than the typical vortex sheet simulation.

1.5 Overview

The next chapter describes the traditional marbling techniques and the physical aspects of the marbling processes. Chapter 3 presents a general review of fluid modeling at three scales, from microscopic to macroscopic levels and introduces mesoscopic modeling. Chapter 4 introduces the Voronoi fluid model based on mesoscale dynamics, and transport and interface models. Chapter 5 presents the digital marbling solution based on an analysis of the traditional techniques, categorizing the results according to the flow Reynolds number. This involves the applications of the numerical fluid model described in Chapter 4. This chapter also introduces the vortex sheet model and the multiscale modeling of fluctuating/turbulent flows using the Voronoi fluid model. Finally, Chapter 6 presents a summary of the proposed methods and results and suggests possibilities for future work.

Chapter 2

Traditional Marbling

2.1 The Basic Technique

The conventional form of marbling, also known as "Ebru" (cloud art), originated in Turkey in 15th century as a form of figurative painting. It then spread to Europe where it was mostly used in bookbinding. The basic marbling technique involves dropping pigments on the surface of a liquid, and manipulating the floating paint with a brush or a comb. The painting created on the surface is then transferred to a sheet of paper. The background liquid, which is called "size", is a gelatinous solution obtained by thickening water with various plants. Size is placed in a tray and the colors are sprinkled or dropped on the surface of the size, with a drop brush, which could be of different types. The stylus or comb which is used to move the colors, can also be of different sizes or shapes, and the pins of the comb can be arranged in different ways according to the desired patterns. Figure 2.1 illustrates the basic technique.

2.2 Materials

When some plants or vegetable based substances that contain gum are mixed with water, a liquid gel is produced. The gum solution that is used as the *size* is prepared in this way. Gum-traganath, cellulose (wallpaper paste), carregeen-moss (seaweed product), methocel are some of the materials used to aid the viscosity of the size. When size is more viscous, it is easier to control the paint in which case the movement is slow, otherwise the flow is more dynamic.

There are a variety of choices for the colors. In the early days marblers prepared them by grinding the pigments themselves. Today many manufactured paints are available; acrylics, watercolors, oil colors or calligraphy inks can be used depending on the marbling style. Some of these need to be diluted with water. Each pigment is obtained from different materials in nature (woods, mercury, minerals or plants etc.), thus they have different physical and chemical properties; such as its density, solubility, volatility, brightness, adherence and friction. Colors are usually mixed with a dispersing agent (if it is not already included in the paint as in some new paints specifically formulated for marbling). Usually oxgall and oil are used as a surfactant. When pigments are dropped on the

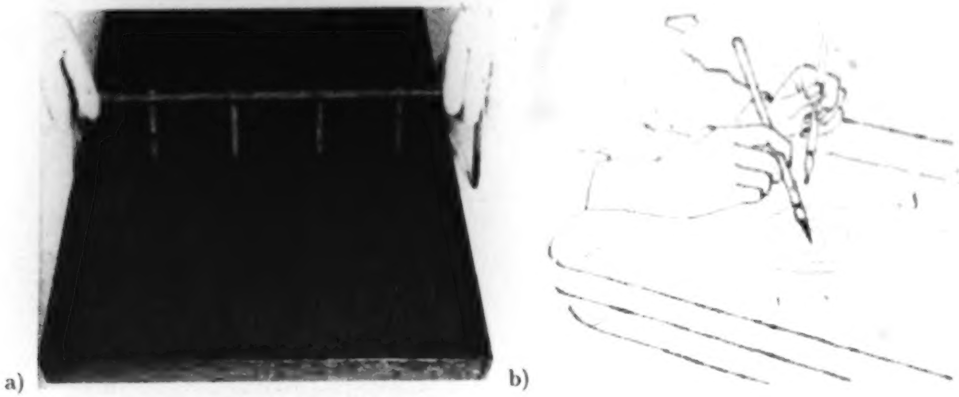


Figure 2.1: Marbling with a) a comb and b) a stylus.

surface of the size, normally they sink due to the gravitational force. Dispersing agent makes the pigment solution float and spread on the surface. It also, keeps the colors from mixing with each other. Mobility of the pigment solution can be adjusted by changing the amount of dispersant used. Apart from the dispersant other additives can be used for different purposes such as to solidify the colors or alter their brightness or adhesion.

Alum is another chemical used in marbling. It is applied on the paper to fix the marbled liquid to the paper, which otherwise would slide off especially when a nonabsorbent paper is used. In my experiments with real marbling, I observed that this is a major difficulty in marbling, especially in low viscosity solutions where most of the details on the marbled surface is lost during the transfer to the paper.

2.3 Physical Processes

The patterns created on the surface is a result of the chemical and physical activities of the materials involved in the process. In the earliest centuries of the craft, marblers kept their techniques as a secret. Even though the first formulas were lost, several studies were done later to explain the chemical and physical properties of the materials [163, 164]. In order to give an insight into the underlying physics, I will briefly explain these processes even though I will not attempt to simulate all the details. Especially the chemical processes are beyond the scope of this thesis.

The pigment and gum solutions in marbling constitute a colloidal system. When particles with a considerably large size are dispersed in a medium then the mixture is called a colloidal suspension. The mixture of solid particles in a liquid form a colloidal suspension when the solid particles have a considerably different size than the medium they are dispersed in. The complex behaviour of these particles have been a subject of interest among the physicists in the last century [163]. The size and shape of the particles, and their interactions determine the behavior of a colloidal system. In marbling size, particles of the gum are dissolved in water, forming a gelatinous liquid. And in colors, the ground pigments make up for the colloids. The properties of these fluids have several effects on the marbling flow.

Inside a liquid, particles are subject to attractive (van der Waals) and repulsive forces equal in all directions; that is, the net force acting on each molecule is zero. However on the surface molecules, there is a resultant force, as the forces from outside and from the interior are not balanced. This force caused by the attraction between similar molecules form surface tension. The dispersing agent, added to the colors, contains hydrophobic (water-fearing) fat and hydrophilic (water-loving) molecules. The water soluble, soap like, molecules in the dispersant cause the fat molecules mix with water by means of micellar solubilization. In other words, they form mixed micelles (molecular aggregates) in which the hydrophobic molecules are attracted to the center of the aggregate while the hydrophilic molecules remain on the outside. In this way, the molecules in the paint solution are attracted to the size (hydrated gel solution) more than they attract each other, and the paint spreads. In addition, the hydrophobic molecules in the dispersant attract the pigment particles (unless they are too big in which case the gravitational force will prevail) inducing a net charge which keeps them from forming agglomerates, as a result they stay hydrated and become subject to the Brownian motion [1] and this keeps them from sinking.

These summarize the two important functions of the surface tension force in marbling, which is created as a result of the interaction between the molecules of pigment solution and the size: spreading and floating the paint on the surface. From a fluid modeling perspective, attraction of the paint to the surface indicates that the flow can be considered two-dimensional.

Other than these, the effect of colloidal systems should be analyzed considering the rheological properties of the fluids, that is, in terms of their deformability characteristics. These depend on the properties of the particles; how the molecules interact with each other and with the molecules of the solvent in which they are suspended and also the concentration of the particle solution.

These events take place at mesoscopic levels, that is at scales larger than the microscopic level and much smaller than the continuum level. Some of the properties of these processes at the mesoscopic level or how they influence the macroscopic level are still not known. This is also true for polymeric liquids which include most of the paints used in marbling as well as the gel-like mixtures. Polymers are usually added in the pigment solution as a binder, that is, to increase the adhesion of the pigments. Natural pigments usually have different adhesion properties depending on the material properties and may not need additives, however among the synthetic paints acrylics usually contain polymers and in oil paints oil act as a binder. Several properties of the polymeric liquids [116] are well defined; for example the polymer molecules in a fluid under convection tend to connect to each other forming chains, which is usually referred to as the "bead-spring" structure. From the visually observable level, this means increased elasticity and more resistance to the flow. The fact that this molecular formation happens at different time and length scales makes the numerical modeling very difficult. When the interaction between different types of molecules, that is, the dynamics of the interface between two fluids is considered, the general hydrodynamic behaviour is towards minimizing the surface free energy based on van der Waals theory [156].

To sum up, in general, together with the particle characteristics, chemical properties of the pigments and the additional materials, the viscosity and surface tension of the size and the pigment solution are the main factors that affect the marbling process. Based on these physical processes let's define the following modeling tasks: providing a hydrodynamic description for the background solution including both Newtonian and colloidal properties, modeling the advection of the paints dispersed in the background, simulation of surface tension effects and interface dynamics considering the rheological properties such as elasticity and mobility to an extent.

---

[1] Brownian motion [64] describes the irregular motion of particles suspended in a liquid, which is mainly caused by the their collisions with the surrounding liquid.

The environmental effects, such as temperature, humidity, and aging also play an important role in preparing the materials. For example, in the real experiments, I observed a significant increase in viscosity in short time periods, this is probably because of some continuous chemical processes (such as bacterial degradation) and the sedimentation of the paint. Surface tension is also affected by the temperature. In general surface tension is reduced with temperature increase.

Apart from the difficulty in preparing the materials, marbling technique also requires a delicate and swift manipulation. Especially in dilute solutions where the flow evolves very fast, the flow is very sensitive to any perturbation; a small change in the geometry of the stylus makes a big difference in the resulting patterns. It is difficult to capture the desired pattern and of course it is not possible to take another similar one after the flow is diverted.

Various experiments [163, 79, 67] have been done and new substances have been discovered over the years to improve the quality of the marbled designs, to overcome the current problems and to create different effects.

2.4 Patterns

The different marbling styles are just variations of the basic technique. They have been invented by modifying the characteristics of the materials involved, the stroke of the artist and the type of tool used. I will classify the patterns according to the physical properties of the size and the paints. The first three types are based on viscosity and the last one relies on surface tension.

Comb marbleized pattern is one of the most well-known marbling designs [109]. The fluid viscosity is very high, the size behaves more like a gel than a liquid. The only motion in the flow is the stretching around the boundary. Figurative patterns can be created using this level of viscosity. Figure 2.2a shows a peacock pattern created by using a comb and Figure 2.2b shows a figurative design designed by a stylus.

In swirl patterns illustrated at Figure 2.3, the viscosity of the size is lower, so that it behaves like a liquid. The swirling patterns are obtained by the complex path of the stylus.

Suminagashi marbling is considered to be the earliest form of marbling. Only water is used as the background liquid, with no additives to increase the viscosity (to thicken the liquid). Turbulent-like patterns which mostly depend on the dynamics of the flow are created with little manipulation. Patterns illustrated at Figure 2.4a are obtained by using a fan, the pattern at Figure 2.4b is a typical Suminagashi design.

To make the shell patterns (Figure 2.5) surface tension of the paints are increased by adding oil or turpentine. The paints do not spread too much and they form droplets. When more viscous paints with high surface tension are put on the surface together with more diluted pigment solutions, thick paints shrink and form veins. The patterns formed are very interesting even without any

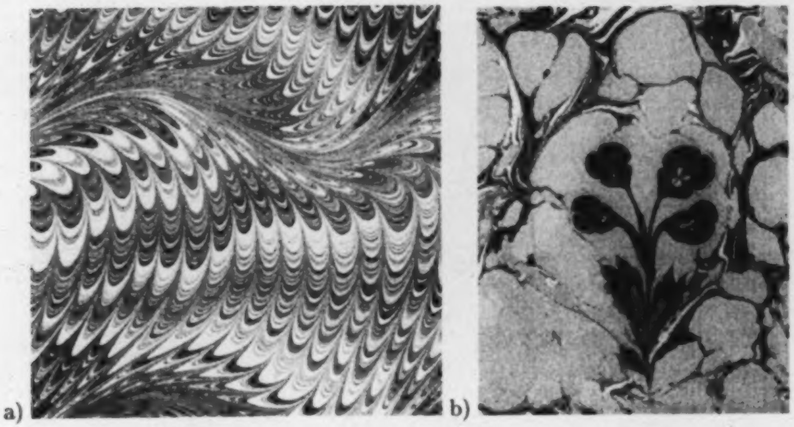


Figure 2.2: a) Peacock pattern [109], b) a figurative design [80].

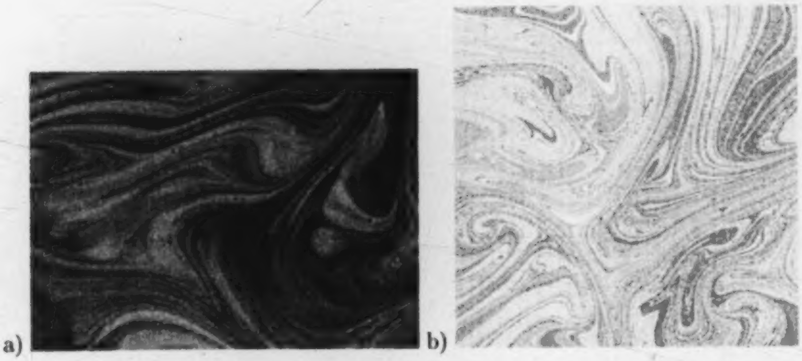


Figure 2.3: Moire patterns a) [81].

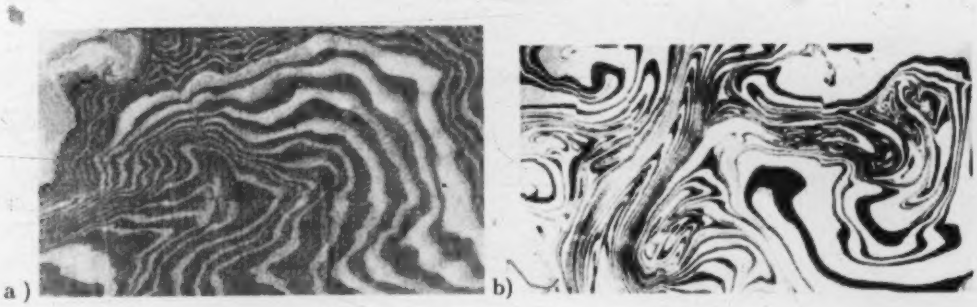


Figure 2.4: Suminagashi designs a) [82].

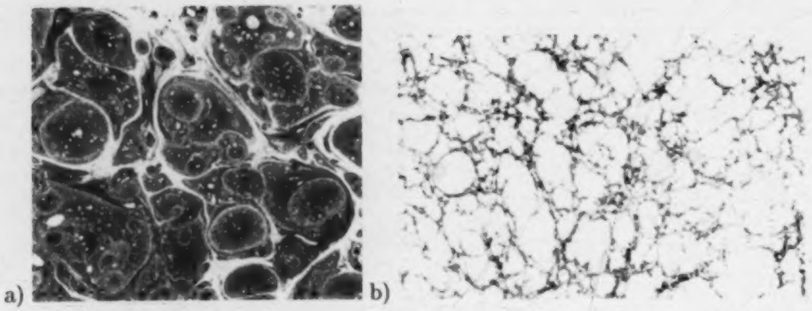


Figure 2.5: a) Shell [86] and b) vein [84] patterns .

manipulation. I obtained the results illustrated at Figure 2.5b, by dropping some acrylics diluted with alcohol on top of thick paint droplets mixed with oil. The veins become finer when more oil is used.

As we have seen in this chapter, marbling involves very complex physical processes. Numerical modeling of these in full detail (and with complete accuracy) is almost impossible with the current knowledge and computer technology. Considering the basic characteristics of the marbling process, simulation of a wide range of flows, from very viscous fluid behaviours (as in comb patterns) to turbulent flows (as in most Suminagashi designs) will be considered in the digital marbling model. Only certain chemical reactions as used in the development of vein patterns will not be considered in this model. In the next chapter, I will present background information on fluid modeling at 3 levels and more specifically on mesoscale dynamics.

Chapter 3

Fluid Modeling at the Mesoscopic Level

Fluid flow can be studied in three levels: microscopic, mesoscopic and macroscopic. In this chapter, first the basics of microscopic and macroscopic levels will be reviewed briefly. Then mesoscopic dynamics and the related studies in CFD will be discussed in more detail. Computer graphics use of fluid mechanics which is mostly based on the macroscopic equations will be reviewed in the following chapters. Mesoscopic modeling has not been addressed in computer graphics to my knowledge. Digital marbling is based on macroscopic and mesoscopic modeling; the level of details involved in microscopic dynamics can be considered irrelevant when average flow behaviours are of concern. Macroscopic equations describe mean flow properties. The effect of mesoscopic properties will be studied in two parts: the first part constitutes the complex behaviours caused by the colloidal properties of the fluids which are assumed to affect the mean flow and the second part includes the interface events which also take place at mesoscopic levels. Interface dynamics is modeled by deterministic equations which will be discussed in the next chapter. The stochastic particle modeling (Dissipative Particle Dynamics) described here constitutes a background for the first part. Note that these mesoscopic forces are still on a small scale compared to visually observable levels, thus their effects on the mean flow will be obtained by averaging (as described in the next chapter.)

3.1 Microscopic Level

At the microscopic level, interactions between individual molecules are studied in terms of Newton's gravitational laws and molecular forces derived from a potential. One might argue that the molecular dynamics should be the most accurate way to simulate fluids. Whether or not the mean flow (macroscopic) properties can be derived from molecular forces is still questionable; there could be molecular motion in a fluid at rest, or molecular movement might remain the same in a flow where there is high momentum transport. Most of the details at the microscopic level are actually not carried to the macroscopic level. Given the computational demands of molecular modeling, this is difficult to find out with today's computer power. In other words, approximating the levels that can be observed with the naked eye from microscopic interactions is computationally unfeasible. The requirements for molecular operations are extensive mainly for two reasons:

1. Very large number of particle interactions have to be computed. Mostly a limited number of particles are considered within a certain range to resolve this problem.

2. The scale of interactions (in time and space) is very small.

For these reasons, this method can be used to model events at the microscopic level only. Due to the expense of computing particle forces, a limited number of particles can be used in the discrete systems, so only simplistic models can be simulated. This has been the case for the few studies that have been proposed to model fluids based on molecular forces in computer graphics [113, 150, 152]. Molecular interaction forces have been used to model the deformation of solids and fluids in [113, 150, 152], however complex fluid behaviours at the continuum level cannot be modeled with these models.

3.2 Macroscopic Level

At the macroscopic level, the fluid domain is divided into control volumes and the mean values of the physical quantities at these macroscopic points are calculated. Fluid motion is described by a set of partial differential equations derived from Newton's conservation laws: conservation of mass, momentum and energy. Newtons first law, conservation of mass, states that the inflow of mass inside a control volume should be equal to the net outflow of mass across the control surface. The second law, conservation of momentum, states that the momentum of a system does not change if there are no external forces acting on the system. In other words, the rate of change of linear momentum of a system is equal to the net force applied to the control volume. The forces acting on a control volume are pressure, viscosity and gravity. Application of these forces to the equation of motion yields the Navier–Stokes equations. The momentum equation in conservation form is given by:

$$\frac{D\mathbf{V}}{Dt} = -\frac{1}{\rho}\boldsymbol{\nabla} p + \nu\nabla^2\mathbf{V}. \tag{3.1}$$

Here $\mathbf{V} = (u, v)$ is the fluid velocity, p indicates the pressure and $\nu = \frac{\eta}{\rho}$ represents the kinematic viscosity with ρ and η representing density and viscosity respectively. Differential terms, $\frac{D}{Dt}$, $\boldsymbol{\nabla}$ and ∇^2, which indicate the temporal derivative and divergence and Laplacian operators, respectively, are given by:

$$\begin{aligned} \frac{D}{Dt} &\equiv \frac{\partial}{\partial t} + u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y}, \\ \boldsymbol{\nabla} &\equiv \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right), \\ \nabla^2 &\equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}. \end{aligned} \tag{3.2}$$

Mass conservation is expressed as:

$$\frac{\partial \rho}{\partial t} = -\nabla\cdot(\rho \mathbf{V}). \tag{3.3}$$

A fluid is incompressible if its density is independent of pressure. The volume of a fluid element in an incompressible fluid does not change in magnitude but may deform, whereas in compressible fluids the magnitude of the particle volume may also change with pressure. Although all materials compress with an increase in pressure to a certain extent, this effect is negligible in most liquids; usually only gasses are considered as compressible. This assumption removes the dependency on density from the equations. When the effect of density change is eliminated (i.e. when density is constant) continuity equation becomes:

$$\nabla\cdot\mathbf{V} = 0. \tag{3.4}$$

The momentum equation consists of the convection, pressure gradient and diffusion terms. Convection term accounts for the transportive property of the fluid. Diffusion equation spreads the velocity vectors with a regular distribution. And the continuity equation can be thought of as a constraint on the equations to obtain fluid pressure. Thus these equations are used to calculate fluid velocity and pressure at each point in a computational space. Although they are considered to be the perfect model for incompressible fluids, the numerical modeling of these equations can be very problematic. Numerical solutions are obtained by discretizing the equations on a domain of control volumes. Due to their nonlinear and diverse characteristics, they may be computationally expensive as well as prone to instabilities and numerical inaccuracies. Especially when the smoothing of the dissipative term is not present or effective, hyperbolic characteristic of the convective term causes propagation of the discontinuities and discretization errors. Numerical solution methods for both mass and momentum equations is a wide area. Numerous methods have been offered for different terms in the equations. In the next chapter, discretization methods will be discussed in more detail.

3.3 Mesoscopic Level

Mesoscale dynamics describes the fluid behaviours between microscopic and macroscopic levels, that is, each particle represents an averaged number of molecules. Complex fluids, such as multiphase micelles and immiscible mixtures and colloidal suspensions, involve complicated flow interactions which occur at different length and time scales. In many cases there is no hydrodynamic description for the flow, or in others the flow characteristics are so complex that it is computationally intractable to include all the details in the fluid model at once. Different approaches have been proposed for modeling complex fluids. These will be categorized as lattice methods, which are based on simulating the motion of particles on a lattice, and off-lattice methods, which are based on statistical descriptions.

3.3.1 Lattice Based Models

Lattice-Gas [58] model is considered the first mesoscopic method. In Lattice-Gas dynamics, particles

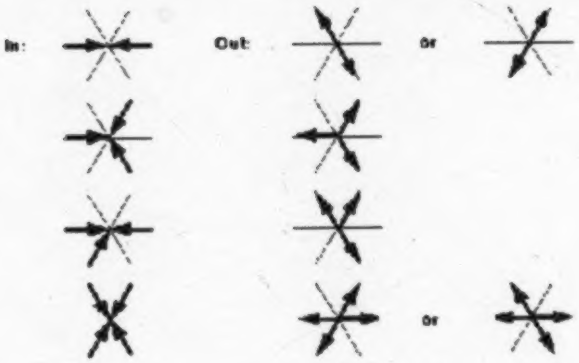


Figure 3.1: FHP lattice gas collision rules [168].

move on a fixed (symmetric) lattice according to some collision rules which are designed to conserve mass and momentum. Several models have been developed [83] based on different lattice structures, Figure 3.1 shows some of the rules for an FHP (Frisch, Hasslacher, Pomeau) lattice gas model [58] which is based on a hexagonal cell structure. Later the discrete nature of the motion in the lattice-gas model was improved in the Lattice-Boltzmann [108] methods by using probability distribution functions, based on the Boltzmann equation, instead of particle collision rules. Lattice based methods are easy to implement, they do not suffer from severe stability restrictions and provide mass conservation. They have been successfully used to model a wide range of flows, from complex fluids to high Reynolds number flows [83, 8, 168]. However, due to a lack of perfect isotropy they are not considered suitable for complex 3D boundary modeling and multiscale simulation. More importantly with regard to numerical modeling, motion is restricted to a lattice and even though they are considered numerically stable it should be emphasized that this is a cellular automata method and the step size is defined by the lattice size. Another issue is that they do not use thermal fluctuations which play an important role in describing the properties of complex fluids according to many physicists.

3.3.2 Stochastic Models

In reality, the equations describing the motion of a mechanical system are deterministic. At both ends of the spectrum, molecular dynamics and Navier–Stokes equations describe the evolution of microscopic and macroscopic quantities, respectively. The physics of the level in between these two extremes cannot be described by classical mechanics, mostly due to a lack of knowledge of the events that occur at this level or due to the complexity of the system. For this reason, properties of this level are described by stochastic processes, assuming that descriptions in full detail are not necessary (or they are not worth the effort). Random fluctuations are added to the system of deterministic equations which is then described by stochastic differential equations. Statistical properties of the fluctuations are then obtained by imposing the fluctuation-dissipation theorem [102] on the equations. Fluctuation-dissipation theorem, determines the ratio of random to viscous forces to reproduce thermodynamic equilibrium.

The Langevin equation [64] is one of the well-known stochastic equations which describes the motion of a Brownian particle. According to Langevin theory, particles suspended in a liquid are subject to hydrodynamic, interparticle and Brownian forces. As a particle starts to move with these forces it changes the state of neighbouring particles and propagates the motion to the surrounding fluid. The dynamics of the particles change according to the particle characteristics, and the Peclet number, which is the ratio of hydrodynamic to Brownian forces. In diluted suspensions Brownian forces dominate whereas in high concentration fluids hydrodynamic forces are more effective. The Langevin equation describes the motion of particles through dependent frictional forces, force fields which are derived from a potential, and fluctuating forces:

$$\frac{d\mathbf{v}}{dt} = -\gamma\mathbf{v} + \sigma\boldsymbol{\xi} + \mathbf{f}_c, \tag{3.5}$$

where $\mathbf{v}$ represents the particle velocity, γ indicates a friction constant, $\boldsymbol{\xi}$ is the random force and σ is the amplitude of the fluctuations, and finally $\mathbf{f}_c$ represents the force derived from a potential.

Brownian dynamics provides an ensemble of the stochastic processes; it describes the motion of colloidal particles suspended in a liquid by frictional and random forces on a time scale larger than the molecular interactions. However in Brownian dynamics, only the number of particles is conserved, thus it is diffusive, not hydrodynamic. In order to obtain the hydrodynamic properties at the macroscopic level, momentum should be conserved as well. Dissipative Particle Dynamics (DPD) [66] was developed to describe the motion of mesoscopic particles according to forces that were formulated to conserve momentum.

3.3.2.1 Dissipative Particle Dynamics

Dissipative particle dynamics simulates the interaction of a number of fluid particles through conservative, dissipative and random forces. It is like a molecular dynamics method but each particle represents a bulk of fluid rather than an individual molecule. It can be considered as a generalization of the Langevin approach. In DPD, the position and velocities of the particles are represented in a continuous space, but time is discrete. Hence, this model is not limited by the drawbacks of a lattice system. The forces are smoothed in a cut-off range around the particles on which they are effective.

Stochastic differential equations that describe the evolution of the positions and velocities of DPD particles are given by [66]:

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_i, \tag{3.6}$$

$$\frac{d\mathbf{v}_i}{dt} = \sum_{i\neq j}\left(\mathbf{F}_{ij}^C + \mathbf{F}_{ij}^D + \mathbf{F}_{ij}^R\right), \tag{3.7}$$

where $\mathbf{v}_i$ and $\mathbf{x}_i$ are the velocity and position vectors of particle i respectively. Note that the momentum equation (Eqn.3.7) assumes unit particle mass. Three components of the particle momentum force are:

$$\mathbf{F}_{ij}^{C} = \begin{cases} a_{ij}(1 - \frac{r_{ij}}{r_c})\,\mathbf{n}_{ij} & 0 < r_{ij} < r_c \\ 0 & r_{ij} > r_c \end{cases}, \quad \mathbf{n}_{ij} = \frac{\mathbf{r}_{ij}}{r_{ij}}, \tag{3.8}$$

$$\mathbf{F}_{ij}^{D} = -\gamma w_D(r_{ij})(\mathbf{n}_{ij}\cdot\mathbf{v}_{ij})\mathbf{n}_{ij}, \tag{3.9}$$

$$\mathbf{F}_{ij}^{R} = \sigma w_R(r_{ij})\mathbf{n}_{ij}\xi_{ij}. \tag{3.10}$$

Here r_{ij} is the distance between particles i and j and $\mathbf{n}_{ij}$ is the unit vector pointing from j to i. The variable r_c is the radius of the sphere defined by the weight functions, which limit the range of forces. The weight functions w_D and w_R smooth out the forces and localize their effect.

The force $\mathbf{F}^C$ denotes the conservative force. It is a repulsive force derived from a central potential whose magnitude depends on the distance between the particles and the maximum repulsion parameter a_{ij}. The dissipative force, $\mathbf{F}^D$, depends on the relative velocity between the particles. Its magnitude is determined by the friction coefficient γ. This viscous force is in the opposite direction to the movement of particle i relative to j. In other words, the dissipative force reduces the momentum of the particles. The random force, $\mathbf{F}^R$, represents the random fluctuations in the system. The variable, ξ_{ij}, indicates a Gaussian stochastic variable with zero mean and unit variance. It should satisfy the symmetry property, $\xi_{ij} = \xi_{ji}$ to ensure momentum conservation. With these forces the momentum equation takes the form:

$$d\mathbf{v}_i = \left[(\mathbf{F}_{ij}^{C} + \mathbf{F}_{ij}^{D})\right] dt + \sigma w_R(r_{ij})\mathbf{n}_{ij} dW_{ij}, \tag{3.11}$$

where dW_{ij} denotes the Wiener process [64] which is discretized as $\xi_{ij}\sqrt{dt}$ according to Ito calculus [64].

In order to satisfy the fluctuation-dissipation theorem [102] the following requirements are imposed on the weight functions, noise amplitude and friction coefficient:

$$\mathbf{F}_{ij}^{R} = \sigma w_R(r_{ij})\mathbf{n}_{ij}\xi_{ij}, \tag{3.12}$$

where $\sigma = (2\gamma k_B T)^{1/2}$ indicates the amplitude of the noise and k_B and T represent the Boltzmann constant and the equilibrium temperature respectively. The following weight function can be used for w_D [121]:

$$w_D = \begin{cases} \frac{3}{\pi r_c^2}(1 - \frac{r}{r_c}) & 0 < r < r_c \\ 0 & r > r_c \end{cases}. \tag{3.13}$$

Figure 3.2 shows the animation frames from Dzwinel and Yuen's study [34] for the simulation of Rayleigh-Taylor mixing of two immiscible fluids at the mesoscopic level by using DPD.

One of the two main problems with DPD is the physical interpretation of the conservative forces. They are not realistic since the fluid pressure can not be described realistically without a thermodynamic description. For a complete hydrodynamic description, where the fluid can be described in terms of the macroscopic fluid parameters such as pressure and viscosity as in continuum

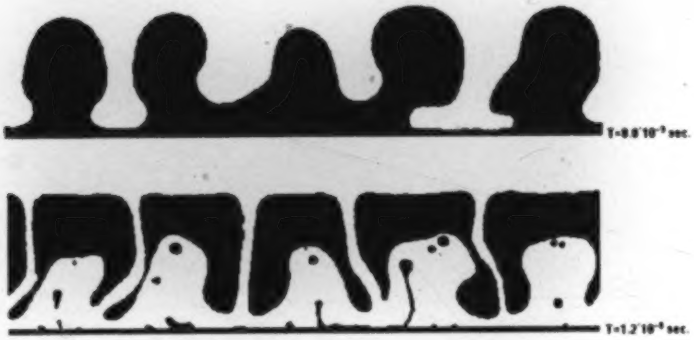


Figure 3.2: Results from the simulation of Rayleigh-Taylor mixing by using DPD [34].

dynamics, energy flow should also be taken into account. The other issue is system scaling. Although mesoscopic modeling allows bigger time and space scales with respect to molecular dynamics, the evolution of the system is still very slow compared to the macroscopic scale. In order to match the macroscopic properties faster the model should be suitable for the approximation of larger scales. The non-central nature of the dissipative force in DPD prevents its use for larger length scales. Attempts to re-scale the system by increasing the particle size [35] showed that there is an upper limit for the particle size after which the system freezes. In other words macroscopic level can not be reached by increasing the time step and/or the particle size in DPD.

Recently there have been some studies to fix these problems [131, 38, 46, 41]. These are based on two different approaches which provide similar results. The first one is a bottom-up approach which applies a local averaging method on the microscopic equations to obtain the fluxes of mass, momentum and energy at the mesoscopic level. The top-down approach is based on discretizing the continuum equations with a finite volume method to obtain the discrete hydrodynamic variables. Each particle is represented by a Voronoi cell in these models. Details of this discretization and their derivations will be discussed in the next chapter.

The dissipative force, in these models is given by $[\mathbf{v}_{ij} + (\mathbf{v}_{ij} \cdot \mathbf{n}_{ij})\,\mathbf{n}_{ij}]$. The only difference here is the first contribution; it is directed along the relative velocities and it dissipates the difference between the velocities. The conservative force in these models is also derived from a potential as in DPD but here this potential is taken to be the internal energy of the system rather than an external field as in DPD. This potential is used to obtain the equation of state which gives pressure a hydrodynamic description. Equation of state of a thermodynamic variable determines the relation between its pressure and temperature in terms of energy and particle volume. It is an initial state which can not be described by the classical thermodynamics, and usually obtained empirically. The pressure and temperature are defined through the *equation of state* of an ideal gas as follows:

$$T = T(E, V) = \frac{2E}{k_B}, \quad P = P(E, V) = \frac{E}{V}, \tag{3.14}$$

where P, T and E indicate pressure, temperature and energy respectively, V represents the cell volume and k_B denotes Boltzmann's constant. Note that, the fluid is not assumed to be an ideal gas here, these equations are used to determine the values in the momentum equation. Each cell is associated with an equation of state for its pressure and energy.

The energy conservation equation is given by:

$$\frac{dE_i}{dt} = \sum_j w(r_{ij}) \left(-k_{th} \frac{T_{ij}}{r_{ij}} + [(\frac{E_i}{V_i} + \frac{E_j}{V_j})\mathbf{v}_{ij} - \mathbf{J}_{ij} \cdot \mathbf{v}_{ij}] + \frac{1}{r_{ij}} \widetilde{T_{ij}} \right). \tag{3.15}$$

The first term on the right, inside the parentheses, represents the heat flux. The parameter k_{th} is the thermal conductivity and T_{ij} is the temperature difference between the particles. The second term, represents the energy due to interparticle forces. Here $\mathbf{J}_{ij}$ denotes the viscous flux due to dissipative and conservative forces. Finally the last term on the right represents the energy caused by the thermal fluctuations, $\widetilde{T_{ij}}$. The properties of the fluctuations in these Voronoi models will be discussed in the next chapter. Note that, the weight function, $w(r_{ij})$, is based on the characteristics of a Voronoi structure and is different from the DPD model; this will also be more clear in the discretization discussion in the next chapter.

DPD method has also been used for modeling interface dynamics of immiscible fluids [121]. In order to simulate these properties I will use a deterministic continuum method based on Cahn-Hilliard theory [15]. I find it more appropriate to use a deterministic model for the interface problem; the small scale forces in the particle representation would make it very expensive to model the interface motion at the macroscopic level, besides due to stochastic nature of the flow only an approximate solution can be obtained. However a DPD type stochastic model will be used to simulate the complex motion and turbulence that might be caused by the complexities at the mesoscopic level, that is colloidal fluctuations. I used an averaging model to obtain the effect of these forces at larger scales. Instead of the particle representation (as in DPD) a mesoscale model which can be obtained by incorporating the mesoscale properties in the macroscopic equations will be used. The basic DPD idea is described here to provide a background for the use of mesoscopic forces in the marbling model which will be described in the next chapter.

Chapter 4

Discrete Fluid Model

This chapter describes the numerical model in two parts. The first part constitutes the solution of fluid equations on a relatively coarse grid to obtain the flow vectors. Incompressible fluid equations including the mesoscopic fluctuations are solved with finite volumes on a Voronoi grid. Fluctuations can be excluded in which case the method represents a basic Navier–Stokes solution that allows only the simulation of laminar flows due to its accuracy level. Stochastic mesoscale forces are incorporated into the momentum equation to simulate turbulent and fluctuating motion caused by mesoscopic complexities. DPD like mesoscopic modeling approaches macroscopic equations at the continuum limit, this has been proved in some studies [131, 38, 46, 41]. I changed the originally Lagrangian and compressible characteristics of this fluid description to develop an incompressible Eulerian model. The foundation of this modification with an analysis on its general connection to the gridless particle discretization is described in section 4.2. Another modification involves the pressure description; while original models are based on a compressible description, in the marbling model pressure is defined according to the incompressibility constraint. Voronoi structure is mainly chosen to adapt the physical properties of the mesoscopic fluid model whose discretization is based on a Voronoi structure. The proposed model allows the simulation of a wide range of flows including deterministic as well as stochastic high Reynolds number effects.

The second part of the simulation takes care of the transport model which describes the material transport throughout the velocity space. This part is implemented on a finer rectangular grid, whose velocity vectors are obtained from the fluid grid through interpolation. Advection equation is solved with an improved (high order) Semi-Lagrangian approach together with a sharp interface model considering the immiscible properties of marbling paints. Interface model is based on Cahn-Hilliard surface energy theory. This model includes the simulation of droplet dynamics as well as other surface and rheological properties. Cahn-Hilliard equation is solved with an implicit spectral method, this method is chosen for its numerical efficiency. Before describing these models in detail, let us first review some of the common spatial discretization techniques related to this study.

4.1 Spatial Discretization Techniques for Hydrodynamic Equations

Numerical methods for partial differential equations in hydrodynamic equations can generally be classified as finite differences, finite elements, finite volumes [47] and meshless [105] and spectral methods [47]. Some of these methods are used in the digital marbling model, others are presented for completeness.

4.1.1 Finite Difference Method

In finite differences [47], equations are represented by samples taken on a Cartesian grid of uniformly or nonuniformly spaced nodes. At each node, the equations are approximated by algebraic expressions based on the differences in adjacent grid points. For example a second order derivative of a variable u at grid point i can be represented as follows:

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_i \simeq \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2}. \tag{4.1}$$

One limitation of this representation is that the implementation of high order differential terms can become complicated, as a result application of implicit integration can be computationally expensive. Another disadvantage is that the time step is limited by the grid size due to its stencil based definition. It also requires special care for unstructured/nonorthogonal meshes. Finite differences have been the most popular method mainly due to their simplicity. Most of the computer graphics research are based on finite differences on structured grids [50, 51, 92, 52].

4.1.2 Finite Element Method

In finite elements [47], the flow domain is divided into a number of triangular elements that are connected via nodes. This method requires the formulation of a complete equation; that is, the individual terms cannot be described separately as in finite differences. Let us take the following equation as a reference:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}. \tag{4.2}$$

Each element is assumed to be piecewise approximated by interpolation functions,

$$u = \sum_{i=1}^{N} u_i N_i, \tag{4.3}$$

where N_i indicate the time-independent shape functions. Then the equations written in terms of these elements are integrated over the entire region:

$$\int \frac{u_i^{n-1} N_i - u_i^n N_i}{\Delta t} d\Omega = \int \frac{\partial^2 (u_i N_i)}{\partial x^2} d\Omega. \tag{4.4}$$

After some arrangements to separate the shape function terms from the unknown values, the equations are written as a matrix system which needs to be solved to obtain the unknown values:

$$\mathbf{K}\mathbf{u} = \mathbf{F}, \tag{4.5}$$

where $\mathbf{K}$ is known as the stiffness matrix and $\mathbf{F}$ represents the load vector which contains all the extra terms which are not related to the nodes u_i. This method is suitable for modeling complex geometries, and it is well known for its good approximation properties. However it requires the solution of a linear system of equations which increases the computational complexity especially when high grid resolutions are required.

4.1.3 Finite Volume Method

In the finite volume method [47], the spatial domain is divided into a finite number of cell volumes. The equations are integrated over the control volumes to obtain the volume averaged variables. Then by applying the divergence theorem, area averaged values are obtained. The equation 4.6 shows the volume and corresponding area averaging for the given equation, $\nabla\cdot\mathbf{j}$, written in divergence form.

$$\int_v \nabla\cdot\mathbf{j}\,dv = \int_A \mathbf{j}\cdot\mathbf{n}\,dA. \tag{4.6}$$

Here $\mathbf{j}$ represents a vector quantity. The averaged solution can then be written in discrete form as:

$$\sum_j (\mathbf{J}_{f_{ij}}\cdot\mathbf{n}_{ij})A_{ij}. \tag{4.7}$$

where $\mathbf{J}_{f_{ij}}$ indicates the flux across the face f_{ij}, $\mathbf{n}_{ij}$ is the face normal and A_{ij} represents the face area.

This approximation is consistent with the conservation property and it does not require a structured mesh. Finite volumes is close to finite elements in the sense that they are both derived from an integral representation.

4.1.4 Gridless Methods

The basic idea in this method is to use local interpolation functions to approximate the partial differential equations (PDE) by a collocated discretization method on a scattered set of nodes. In this way the PDEs are approximated in terms of the nodes without the need for a mesh topology. The main advantage of this method is its easy adaptation to large deformations. For this reason, it is mostly used for problems which are characterized by a continuous change in their geometry (such as free surface flows [105]). In order to understand the general idea behind these methods let us review the Smoothed Particle Hydrodynamics method [114]. The key point of SPH is representing each function as an integral interpolant:

$$F(x) = \int F(x')w(x - x', r)dx', \tag{4.8}$$

where w indicates a weight function, with range r, with the properties:

$$\int w(x - x', r)dx' = 1. \tag{4.9}$$

Given the discrete form of this function,

$$F(x) = \sum_{i=1}^{N} \Delta x_i F(x_i) w(x - x_i, r), \tag{4.10}$$

where Δx_i represents a volume element, its gradient can be approximated by using the gradient of the interpolant (due to integration by parts):

$$\nabla F(x) = \sum_{i=1}^{N} \nabla x_i F(x_i) \nabla w(x - x_i, r). \tag{4.11}$$

The gradient of the weight function can be obtained by ordinary differentiation without any need for differentiation methods which use the relation between grid points such as finite differences. In addition, this method does not require the neighbourhood information for each particle. These make this discretization form very easy. However the main problem with SPH is in discrete form most weight functions do not satisfy the consistency conditions:

$$\sum w(x - x, r)\Delta x_i = 1, \tag{4.12}$$

$$\sum (x - x_i) w(x - x_i, r)\Delta x_i = 0. \tag{4.13}$$

Consequently, the interpolatory condition does not hold at data points, $F_0 \neq F(x_o)$. This is also caused by the ambiguity in assigning a discrete form to the volume element Δx_i which represents an average area around the particle. Consistency can be ensured either by minimizing the difference between the real and approximated values or by using Taylors polynomials for the weight functions (as in meshless methods such as Moving Least Squares, Reproducing Kernel, HP-clouds and Partition of Unity). In corrective SPH [105] the kernel function is modified by multiplying the weights with a correction function, then the corrected weights are obtained by solving a system of equations arising from enforcing the consistency conditions. A more general representation of the meshless idea based on a moving least square approximation can be given by first representing the function in terms of its weight coefficients:

$$F(x) = \sum_{i=1}^{m} q_i(x) b_i(x). \tag{4.14}$$

Here q_i are monomial functions of length m and b_i indicate their coefficients which are obtained from a weighted least squares fit by minimizing the difference between the real and approximated values:

$$\varepsilon = \sum_{j=1}^{N} w(x - x_j) \sum_{i} \left[q_i(x_j) b_i - F(x_j)\right]^2, \tag{4.15}$$

where $w(x - x_j)$ is the weight function. The system of equations for the minimization is formulated by setting the derivatives of ε with respect to the coefficients to zero. The coefficients are then obtained by solving the resulting matrix system:

$$\mathbf{b}(x) = [\mathbf{Q}^T\mathbf{WPQ}]^{-1}\mathbf{Q}^T\mathbf{WF}. \tag{4.16}$$

Substituting this into Eqn 4.14 gives the following representation for the approximation of the function:

$$F(x) = \sum_{i=1}^{N} \Psi_i(x) f_i, \tag{4.17}$$

where Ψ are the basis functions defined by $\Psi = \mathbf{Q}^T[\mathbf{Q}^T\mathbf{WQ}]^{-1}\mathbf{Q}^T\mathbf{W}$. When the order of the basis functions is zero, they are equivalent to Shepard's interpolants. Also SPH is equivalent to Shepard's interpolation [132] if Shepard's weights are used. In higher orders (as in meshless-Galerkin approaches [105]), these weight functions are obtained by solving the above matrix system. In this case, local particle distribution should be known to solve the equations. As a result, the computation is no longer simple as in SPH. The main advantage of meshless methods at this point is their flexibility in spatial discretization. Note that, high order formulation of this system is in a way similar to the FEM formulation. (Moving Least Squares meshless approximation can be considered as a generalization of FEM [117].)

We have seen that meshless methods are based on data fitting ideas to approximate the partial differential equations. Interpolation and approximation requirements are usually different. In approximation accuracy has higher priority, while in interpolation smoothness and continuity also needs to be considered. For example, higher order basis functions are advantageous in interpolating arbitrary data, however they might cause difficulties in approximating discontinuities.

In computer graphics, several SPH based models have been proposed. In [31, 144] SPH was used to model deformable surfaces and lava flows. SPH was originally designed for compressible fluids and most of the first models in graphics were based on the compressible description. Lately [124] introduced a more advanced SPH discretization including the incompressibility solution, allowing the simulation of a wider range of fluids with more dynamic properties. FEM like gridless approximation, that is higher order meshless approaches have not been considered in graphics to my knowledge.

4.1.5 Spectral Methods

Spectral methods [47] are based on a similar formulation as FEM; each variable is represented in a variational form in terms of basis functions, which are then substituted into the equations to obtain a solution described by these shape functions. In spectral methods the basis functions are defined as global smoothed functions such as Fourier series transformations by which the derivatives can be described using trigonometric functions. Note that this discretization is considered more efficient on regular grid structures. Given the reference equation,

$$\frac{\partial m}{\partial t} = \frac{\partial^2 m}{\partial x^2}, \tag{4.18}$$

the variables can be represented by discrete Fourier transform as follows,

$$m(x) = \sum_{k=1}^{N} \hat{m}_k e^{i2\pi kx/N}, \quad x = 1, .., N$$

where $\hat{m}_k$ denote Fourier coefficients and j indicates the wavenumber. Derivatives in Fourier space can easily be represented as a multiplication of the Fourier coefficient with the wave number and the imaginary unit,

$$F\left\{\frac{dm}{dx}\right\} = \sum_{k=1}^{N} \hat{m}_k e^{i2\pi kx/N} = ij2\pi\hat{m}_k,$$

and,

$$F\left\{\frac{d^2m}{dx^2}\right\} = -(j2\pi)^2\hat{m}_k.$$

This is one of the main advantages of working in Fourier space, where high order derivatives can be implemented easily and with high order of accuracy (especially in comparison with the finite difference approximation). In addition, since integration is represented by multiplication in Fourier space, implicit formulation for time integration can be less problematic. Once the differentiation is solved in Fourier space then the real values are obtained from the inverse transform. Sine or cosine transformations can also be used instead of Fourier transforms. However these assume a uniform grid structure.

To sum up, spectral methods provide fast and accurate solutions but they are limited to regular grids and can have difficulties in handling discontinuous data. Another disadvantage of these methods is that they don't support local boundary implementation.

4.2 Discretization of Navier–Stokes Equations In Computer Graphics: A General Overview

Most of the fluid studies, in computer graphics, are based on the solution of continuum (Navier–Stokes) equations. One of the first fluid models addressing the solution of Navier–Stokes equations was Kass and Miller's [92] model which used an approximation of the shallow water equations to create fast animations. In their method, a grid of height fields on which the 2D equations were applied to represent the surface of the flow. However, their model cannot simulate the basic rotational fluid motions. Chen et al. [18] solved full Navier–Stokes equations in 2D and obtained the third dimension by increasing the surface height according to the pressure values. They proposed adjusting the flow parameters to achieve stability. A more realistic 3D model was presented by Foster et al. [50], who also used a finite-difference approximation to the Navier–Stokes equations. Later they proposed a method for controlling the fluid animation with approximate control functions [51]. They achieved stability by controlling the fluid viscosity. Also [162] solved the compressible fluid equations. This method is suitable for modeling shock waves, instabilities and other compressible effects.

All of these models are based on explicit finite differences and are limited by stability conditions. This is due to the fact that, explicit integration greatly narrows the stability region of the equations. On the other hand, fully implicit integration is not always the best, or at least the most practical, solution mostly due to nonlinear convective terms which complicate the matrix inversion. Operator splitting is an alternative solution to this problem as it allows the application of different methods for different terms in the equations. This approach decreases the accuracy of the solution however this is not an issue in computer graphics in most cases. Stam introduced such a fractional step method in graphics [140] for the solution of Navier–Stokes; they used an unconditionally stable semi-Lagrangian method for the convection term, and an implicit finite difference method for the diffusion term. This method eliminates the stability restrictions and has been used in many fluid applications in computer graphics [43, 141, 120, 53, 37].

Gridless particle discretization of Navier–Stokes which includes only SPH approach is mostly used for modeling free surface flows will be discussed in (Chapter 4, Section 4.6).

4.3 Voronoi Fluid Model

This section describes the Voronoi fluid model; this involves the adaptation of the mesoscopic properties and discretization of the incompressible fluid equations.

4.3.1 Momentum Equation With Mesoscale Fluctuations

In chapter 3, DPD type mesoscopic modeling idea has been introduced. In this section, first I will present an analysis on the relations between the gridless and finite volume methods and DPD type mesoscale particle models, then I will discuss the modifications I made to the original mesoscale model to adapt it to the marbling simulation.

First let's examine the two mesoscale approaches (mentioned in Chapter 3.) which are based on derivations from microscopic and macroscopic levels. In both of these models each particle is represented by a Voronoi cell and the discretization uses the same sampling function:

$$w_i(x) = \frac{F(x - x_i)}{\sum_j F(x - x_j)}, \tag{4.19}$$

where x_i and x_j are the positions of two neighbouring cells. Note that, this function is the integral interpolant used in SPH.

In derivation of mesoscale equations from the microscopic forces [46], first mesoscopic variables are described in terms of the underlying microscopic variables by using a sampling function as in integral interpolant formalism of gridless methods. Equations for mass, momentum and energy of mesoscale particles are obtained in a similar way. In this section, I will analyze the momentum formulation. Momentum of a mesoscopic particle written in this form is given by:

$$\mathbf{P}_i = \sum_j w_i(x_j)\, m\, \mathbf{v}_j, \tag{4.20}$$

where m and $\mathbf{v}$ indicate microscopic particle's mass and velocity respectively. It is assumed that particles have the same mass.

Derivative of this equation gives the momentum evolution for the mesoscopic variables:

$$\frac{d\mathbf{P}_i}{dt} = \sum_j (\dot{w}_i(x_j)\, m\, \mathbf{v}_j + w_i(x_j)\, m\, \dot{\mathbf{v}}_j). \tag{4.21}$$

After some manipulations, this equation is averaged by using ensemble averages to describe the stresses in terms of the mesoscale variables, instead of the microscopic values and more importantly to represent the mesoscopic variables in terms of average and fluctuating parts. This coarse-graining technique which is related to Fokker-Plank formalism [64] is used to split variables into average and fluctuating components. Accordingly fluctuations are represented by,

$$\tilde{x} = x - \langle x \rangle \,. \tag{4.22}$$

In fact this is just a form of expression to introduce the stochastic components into the system. After this averaging process, the momentum equation including the mesoscale fluctuations is written as:

$$\frac{d\mathbf{P}_i}{dt} = \sum_k w_{ik}(x_k)(\langle \mathbf{J}_k \rangle \cdot \mathbf{r}_{ik} + \tilde{\mathbf{R}}_{ik}). \tag{4.23}$$

Note that, here a term which describes the effect of changes in Voronoi face orientation as the Voronoi centers move is neglected for simplicity. This term is usually referred to as the *boundary twisting* term in these studies [46], and it is expected to have a small effect. In Eqn. 4.23, $\mathbf{r}_{ik}$ indicates the vector pointing from i to k and w_{ik} indicates the sampling function defined in the overlapping region between cells i and k. As a result of integrating this function over space Voronoi face areas are obtained. The equations in the final form are represented in terms of these Voronoi elements. The fluctuating component $\tilde{\mathbf{R}}_{ij}$ is assumed to be Markovian and $\mathbf{J}_k$ represents the momentum flux tensor.

In derivation from the macroscopic level [131], Serrano et.al. used the same smoothing function on a Voronoi structure to sample the equations at the continuum level. This function approaches Voronoi characteristic function as the width goes to zero, and the smoothed sampling gives the finite volume discretization. Fluctuations are included in the equation as a random stress.

Similar models are obtained from both approaches with only difference being the interpretation of individual terms in momentum and heat flux tensors. We have also seen that the gridless discretization of SPH type would lead to the FV formulation on a Voronoi structure. Given these, a general representation for the momentum flux can be given in divergence form including the mesoscale fluctuations as follows:

$$\frac{\partial \mathbf{P}}{\partial t} = \nabla \cdot (\rho \mathbf{u}\mathbf{u} + p\mathbf{I} - \mathbf{\Pi} + \tilde{\mathbf{\Pi}}), \tag{4.24}$$

where the terms from left to right represent convection, pressure, viscous and random flux tensors.

Original mesoscale models are mostly based on a compressible flow description where pressure is described by an *equation of state* as discussed in Chapter 3. However, considering the incompressible properties of most marbling fluids, I defined the pressure term based on the continuity constraint in the marbling model to enforce the incompressibility condition.

Another point is the description of the particle motion. Fluid flow can be described in terms of Eulerian or Lagrangian motions. In gridless and mesoscale models the particle motion is defined as Lagrangian. In Voronoi based mesoscale models [131, 46] the centers of the Voronoi cells follow the flow. In fact, except for the Voronoi based model derived from the microscopic level (in which the derivation originates from microscopic particle forces) the motion in these models is assumed to be Lagrangian. The convective term is not included in the momentum stress as it is realized in the motion of the particles.

The choice between Eulerian and Lagrangian descriptions depends on the characteristics of the flow. Lagrangian description is advantageous in simulating free surface flows since following the fluid elements is trivial. Eulerian approach requires additional mathematical formulation to model this tracking. In Lagrangian description one doesn't have to deal with numerical difficulties caused by nonlinear convective terms, however this approach has some difficulties too. Adaptive mesh modeling is not suitable for very complex surfaces, and it is computationally inefficient. In particle modeling, keeping track of particle neighbours and obtaining a smooth continuous surface between the particles can be problematic. Relative motion of the particles needs to be controlled; sometimes this is solved by introducing extra particles when the particles move further apart and removing some of the particles that occupy the same area. (See Section 4.6 for more discussion on interface modeling with hybrid particle level set methods [37, 53].) In mesoscale models Lagrangian description is chosen mostly to be able to model the interaction between multiple phases at different scales. However I find the Eulerian description more convenient in the marbling model since the fluctuations will be used only for their effect on the mean flow. In addition Eulerian description requires less computational effort considering the computational time that would take to update the Voronoi structure at each step. Free surface flows are not related to marbling type fluids however for the interface tracking problem I will suggest a solution which does not need particle modeling (Section 4.6).

4.3.2 Voronoi Grid Structure

In order to preserve the physical properties of the original mesoscale model, marbling model is also based on a Voronoi structure; grid points are represented as Voronoi cells. A Voronoi diagram of a set of points divides the space into volume elements around each point, with each element being the region of space that is closer to this point than to any other point. The Voronoi characteristic function has the partition of unity property:

$$\sum_i w(x - x_i) = 1, \tag{4.25}$$

where i indicates summation over the neighbour cells. This means that, space is divided into non-overlapping convex regions. Interconnection of the nodes produces a Delaunay triangulation, that is, the dual of Voronoi tessellation. Delaunay triangulation consists of non-overlapping triangles; in other words, it provides a unique triangulation such that no points are enclosed within the circumcircle of each triangle. The center of each circumcircle constitutes a Voronoi vertex (Figure 4.1). The circumcenters of all the neighbouring triangles of each node, that is, the triangles that contain that node, define the vertices of its Voronoi region. The orthogonality property indicates that each

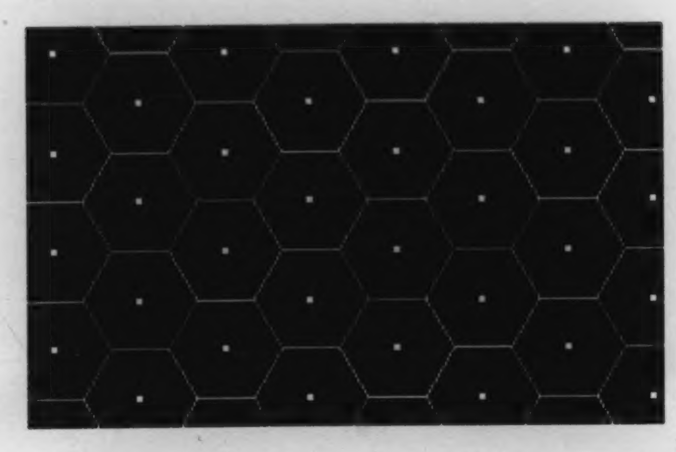


Figure 4.1: Circumcircles that define Voronoi regions.

Voronoi edge is a perpendicular bisector to the corresponding triangle edge; it bisects the triangle edge halfway from each node.

Point distribution is defined quite uniformly on the Voronoi fluid grid, Figure 4.2 shows the Voronoi diagram and the corresponding Delaunay triangulation for a typical grid structure with 210 cells. I tried to avoid highly distorted triangles for two reasons: first to prevent instabilities that might be caused by very small Voronoi edges, second to prevent instability and inaccuracy that might be caused by obtuse triangles. The second case occurs when the circumcenter of a triangle does not lie inside the triangle (Figure 4.3). This means that, there may not be a perpendicular Voronoi edge corresponding to each triangle edge although the reverse condition is always true. This produces an inaccurate finite volume discretization, if not unstable.

For its several useful properties, Voronoi/Delaunay tessellations have been widely used for mesh generation in computer graphics. They are also advantageous in numerical solution of the fluid equations in the marbling model for several aspects:

- Orthogonality property provides a more accurate and simpler approximation of the discrete operators since estimation of tangential terms is not required. In other words, the interpolation error caused by approximating the non-orthogonal terms is removed.

- The regularity and symmetry of the point distribution (which constitutes hexagonal cells in most of fluid domain) together with the orthogonality property improves the accuracy and stability conditions. It also allows faster convergence in the Poisson solution, due to diagonal domination.

- Capability of local grid refining (with $O(N)$ point insertion complexity) makes the Voronoi structure suitable for external boundary treatment. It also allows easy adaptation of the fluid model to complex geometries.

- Non-overlapping space division conforms to the consistency condition of finite volume discretization. Consistency of an equation indicates the development of the approximation as

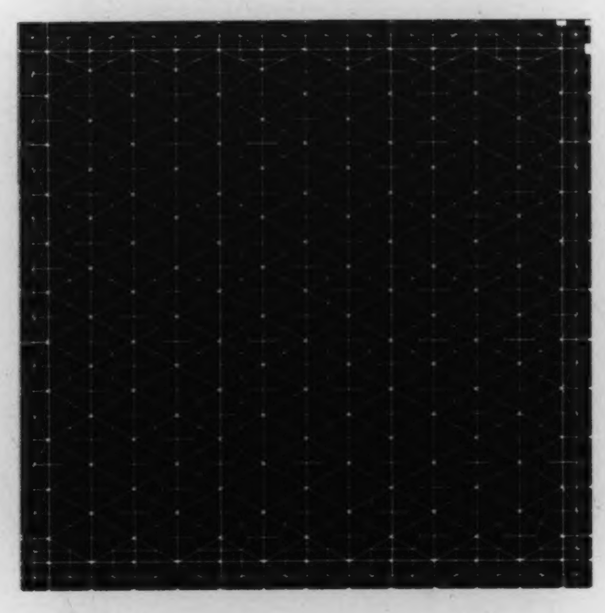


Figure 4.2: Voronoi grid with the corresponding Delaunay connections.

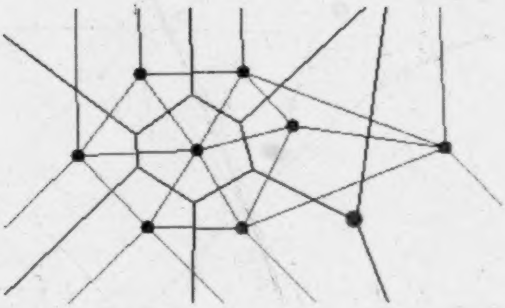


Figure 4.3: Inaccuracy in finite volume cell structure caused by an obtuse triangle.

the cell area approaches zero. Also together with orthogonality and symmetry this improves conservation property.

- Discretization on the Voronoi structure can easily be applied to surfaces in 3D.
- Regular hexagonal cell structure which covers the majority of the fluid grid in the marbling model improves fluid symmetry conditions.

4.3.3 Spatial Discretization of the Momentum Equation

Finite volume discretization based on the divergence form of the equations guarantees local and global conservation depending on the convergence properties of the numerical solution. Other options for discretizing the fluid equations on a Voronoi grid with higher accuracy could be FEM or meshless-Galerkin methods based on a Voronoi structure (such as Natural Element Method [146]). However these methods, as mentioned earlier are computationally expensive and they do not necessarily provide better stability conditions. Besides FEM provides only global conservation.

Now let us review the finite volume approximation of the momentum equation. When constant density is assumed the momentum equation (Eqn. 4.24) reduces to:

$$\frac{d\mathbf{u}}{dt} = \nabla\cdot(\mathbf{u}\mathbf{u} + P\mathbf{1} - \mathbf{\Pi} + \tilde{\mathbf{\Pi}}). \tag{4.26}$$

Finite volume discretization of the momentum equation in 2D is obtained by first integrating Eqn. 4.26 over the control volumes:

$$\frac{\partial}{\partial t}\int_{\Omega} \mathbf{u}\,dA = \int_{\Omega} \nabla\cdot(\mathbf{u}\mathbf{u} + P\mathbf{1} - \mathbf{\Pi} + \tilde{\mathbf{\Pi}})\,dA, \tag{4.27}$$

where Ω indicates the cell boundary. Then by applying the divergence theorem, the following discrete form can be obtained for the momentum equation in terms of fluxes across the cell faces:

$$A_i \frac{\partial \mathbf{u}}{\partial t} = \sum_{j=1}^{n} L_{ij} \left(\mathbf{F}_{C_{ij}} + \mathbf{F}_{P_{ij}} - \mathbf{F}_{D_{ij}} + \mathbf{F}_{R_{ij}}\right), \tag{4.28}$$

where L_{ij} indicates the Voronoi edge length, $\mathbf{F}_C$ and $\mathbf{F}_D$ represent the convective and viscous fluxes respectively, $\mathbf{F}_P$ indicates the pressure gradient and $\mathbf{F}_R$ represents the momentum fluctuations between cells i and j (Figure 4.4).

The flow variables, pressure and velocity, are located at the cell centers. This is usually referred to as collocated or non-staggered discretization. In staggered methods, pressure is defined at the cell centers while velocity is defined on the cell faces; in this case the face normal velocity is the primary unknown in the system of equations. Staggered representation is not subject to high pressure oscillations caused by velocity-pressure decoupling which occurs in a collocated grid. (See Section 4.3.5.3 for a more detailed discussion of this problem.) On the other hand non-staggered representation provides higher accuracy, simpler implementation and uses less memory. Next, discretization of basic operators and individual terms in the momentum flux will be presented. Each term is designed to obtain an optimum discretization on the cell structure illustrated in Figure 4.4.

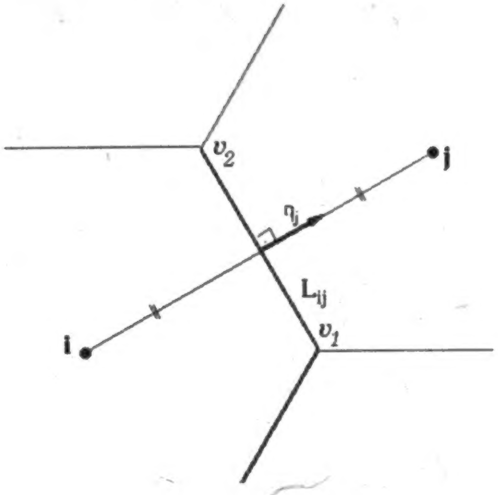


Figure 4.4: Voronoi cell structure.

Divergence:

$$\int_{\boldsymbol{\Omega}} \boldsymbol{\nabla}\cdot\mathbf{u}\, dA = \oint_{\partial\boldsymbol{\Omega}} \mathbf{u}\cdot\mathbf{n}\, dl \Rightarrow$$

$$\sum_j \oint_{\partial\Omega_j} \mathbf{u}\cdot\mathbf{n}\, dl = \sum_j (\mathbf{u}_{\mathbf{f}_{ij}}\cdot\mathbf{n}_{ij}) L_{ij} \tag{4.29}$$

Here, $\mathbf{u}_{f_{ij}}$ represents the approximation of velocity at the cell face f_{ij}, and $\mathbf{n}_{ij}$ denotes the unit normal to the face.

Pressure Gradient:

$$\int_{\boldsymbol{\Omega}} \boldsymbol{\nabla} P dA = \oint_{\partial\boldsymbol{\Omega}} P\mathbf{n} dl \Longrightarrow$$

$$\mathbf{F}_{p_{ij}} = \sum_j \oint_{\partial\Omega_j} P\,\mathbf{n}\, dl = \sum_j (P_j - P_i)\mathbf{n}_{ij} L_{ij} \tag{4.30}$$

Convective flux:

$$\int_{\boldsymbol{\Omega}} (\mathbf{u}\cdot\boldsymbol{\nabla})\mathbf{u} dA = \oint_{\partial\boldsymbol{\Omega}} (\mathbf{u}\cdot\mathbf{n})\mathbf{u} dl \Longrightarrow$$

$$\sum_j \oint_{\partial\Omega_j} (\mathbf{u}\cdot\mathbf{n})\mathbf{u} dl = \sum_j (\mathbf{u}_{\mathbf{f}_{ij}}\cdot\mathbf{n}_{ij})\mathbf{u}^*_{ij} L_{ij} = \sum_j \Phi_{c_{ij}} \mathbf{u}^*_{ij} L_{ij} \tag{4.31}$$

The cell face velocity can also be approximated, probably more accurately, by using the interpolated values at the cell edges in discretization of the line integral over the face as follows:

$$\mathbf{u}_{f_{ij}} = \frac{\mathbf{u}_{v1} + \mathbf{u}_{v2}}{2}. \tag{4.32}$$

This approximation with endpoint interpolation is less diffusive, however as it relies on the interpolation shape function conservation can be poor and the resulting vectors can be asymmetric. For this reason, I mostly used the simpler approximation:

$$\mathbf{u}_{f_{ij}} = \frac{\mathbf{u}_i + \mathbf{u}_j}{2}. \tag{4.33}$$

This is also compatible with the formulation used in the incompressibility solution. In Eqn. 4.31, $\mathbf{u}^*$ represents the upwind element, which is defined as:

$$\mathbf{u}^*_{ij} = 0.5\{(\mathbf{u}_i + \mathbf{u}_j) - \mathrm{sign}(\Phi_{ij})(\mathbf{u}_j - \mathbf{u}_i)\}. \tag{4.34}$$

The whole convective flux can then be written as:

$$\mathbf{F}_{c_{ij}} = \sum_j \{\mathbf{u}_i \max(\Phi_{ij}, 0) + \mathbf{u}_j \max(-\Phi_{ij}, 0)\} L_{ij}. \tag{4.35}$$

The main drawback of the first order upwind scheme is its diffusivity which becomes more apparent when small time steps are taken. Formulations of higher order methods such as ENO [133] or QUICK [47], on unstructured grids are some of the alternatives for discretization of the convective flux. However they require more complex formulations on unstructured grids and they have more restrictive stability conditions. Artificial diffusion method which uses high order Laplacian smoothing in approximating the convective and viscous fluxes could also be an option. Another alternative is to use a Semi-Lagrangian method for convection to allow larger time steps. In this case convection needs to be solved separately by using operator splitting which would decrease the accuracy of the whole system. In addition stability limits of the rest of the equation should be consistent with advection. In other words, discretization of diffusion and other components should be comparable to the unconditional stability of the Semi-Lagrangian method. Operator splitting and Semi-Lagrangian discretization will be discussed in more detail in Section 4.6.

Viscous flux:

$$\int_\Omega \nu \nabla\cdot(\nabla\mathbf{u} + \nabla\mathbf{u}^\mathbf{T})\, dA = \oint_{\partial\Omega} (\nabla\mathbf{u} + \nabla\mathbf{u}^\mathbf{T})\, \mathbf{n}\, dl \Longrightarrow$$

$$\mathbf{F}_{D_{ij}} = \sum_j (\mathbf{D}_{f_{ij}} \mathbf{n}_{\mathbf{ij}}) L_{ij} \tag{4.36}$$

where $\mathbf{D}_{f_{ij}} = (\nabla\mathbf{u})_{\mathbf{ij}} + (\nabla\mathbf{u})^\mathbf{T}_{\mathbf{ij}}$. In gradient term partial derivatives are discretized as follows,

$$\frac{\partial u}{\partial x} = \frac{(u_j - u_i)}{r_{ij}} n_{ij_x}, \quad \frac{\partial u}{\partial y} = \frac{(v_j - v_i)}{r_{ij}} n_{ij_y}. \tag{4.37}$$

Alternatively, and maybe more accurately, the gradient term can be represented in terms of normal, $\frac{\partial u}{\partial n_{ij}}$, and tangential components, $\frac{\partial u}{\partial t_{ij}}$, using the interpolated endpoint values:

$$\frac{\partial u_x}{\partial n_{ij}} = \frac{(u_{x_j} - u_{x_i})}{r_{ij}} n_{ij_x}, \qquad \frac{\partial u_y}{\partial t_{ij}} = -\frac{(u_{y_{v2}} - u_{y_{v1}})}{L_{ij}} n_{ij_y}. \tag{4.38}$$

However as in convection this approximation produced asymmetry due to the interpolation error. In order to achieve higher accuracy, using artificial dissipation method [110] or Natural Element Method could be some of the alternatives. Diffusion discretization is more susceptible to instabilities than convection. I applied kriging smoothing to stabilize the system.

Fluctuations: Fluctuations are included in the finite volume discretization as a random stress term:

$$\int_{\mathbf{\Omega}} \nabla \cdot \tilde{\mathbf{\Pi}} \, dA = \oint_{\partial \mathbf{\Omega}} \tilde{\mathbf{\Pi}} \, \mathbf{n} \, dl \implies$$

$$\mathbf{F}_{R_{ij}} = \sum_j \widetilde{\mathbf{R}_{ij}} L_{ij} \tag{4.39}$$

One might also think each flow variable as decomposed into average and fluctuating components:

$$\mathbf{u} = \overline{\mathbf{u}} + \mathbf{u}'. \tag{4.40}$$

Substituting these variables into the momentum equation and separating the terms which include the random components after the finite volume integration will produce an extra term in the momentum equation which only depends on the stochastic variables.

The random stress can be represented either by a one component random variable using the same number in both directions (as in early DPD methods [66, 121]) or by introducing a random stress as proposed in [131, 38, 46]. In case of random of vectors, components of the vector or the random matrix are designed to comply with certain correlations to ensure symmetry properties. As mentioned previously, in order to ensure conservation random forces need to satisfy the symmetry condition, $\widetilde{\mathbf{R}_{ij}} = \widetilde{\mathbf{R}_{ji}}$. In the marbling model, I used a random vector with two components as follows:

$$\tilde{\mathbf{R}}_{ij} = \sigma_{ij\|} \boldsymbol{\xi}_{ij\|} + \sigma_{ij\perp} \boldsymbol{\xi}_{ij\perp}, \tag{4.41}$$

where $\boldsymbol{\xi}_{ij\|}$ and $\boldsymbol{\xi}_{ij\perp}$represent the two components of the fluctuations in normal and tangential directions to the cell face. They are defined as uncorrelated Gaussian random variables with zero mean and unit variance. Since symmetry is enforced during the implementation (as discussed in Chapter 5), one doesn't have to deal with correlated random number generation in this case. A vector force, rather than a single component random variable, is chosen to increase the randomness.

Statistical properties of the fluctuations are obtained from the fluctuation-dissipation theorem, which is based on balancing the random and viscous forces in accordance with the equilibrium theory. Fluctuation-dissipation condition can be enforced through the Fokker-Plank or GENERIC formalisms. (GENERIC [131] is a framework for non-equilibrium thermodynamics, in which the equations are decomposed into reversible and irreversible parts, in a way which is consistent with the equilibrium distribution of the variables.) The underlying physics in both of these approaches is not different and the properties of the fluctuations obtained from both are similar. Details of this theory is beyond the scope of this thesis however basically in Fokker-Plank formalism this relation is obtained by substituting Einstein's equilibrium distribution,

$$\rho^{eq} \sim \exp(-\frac{\sum M_i/m_i}{k_B T}), \tag{4.42}$$

into the Fokker-Planck equation:

$$\frac{\partial \rho}{\partial t} = \mathcal{L}_C \rho + \mathcal{L}_D \rho + \mathcal{L}_R \rho. \tag{4.43}$$

The operator $\mathcal{L}$ is the Lioville operator which gives the derivatives according to particle momentums, $\mathcal{L} \sim \frac{\partial}{\partial P}$. $\mathcal{L}_C$, $\mathcal{L}_D$ and $\mathcal{L}_R$ represent the conservative, dissipative and random terms, respectively. At equilibrium, $\frac{\partial \rho^{eq}}{\partial t} = 0$, conservative term is assumed to vanish leaving the relations between dissipative and random forces. These are generally defined as:

$$\sigma^2_{ij\|} = 2\sigma^2_{ij\perp} = \frac{4\gamma k_B \overline{T_{ij}}}{r_{ji}}, \quad \overline{T_{ij}} = \frac{2T_i T_j}{T_i + T_j}. \tag{4.44}$$

Here k_B indicates the Boltzmann constant and T_i is the temperature, I assigned a constant value for the temperature at each point in my experiments but it can also be altered for different effects. Note that, magnitude of the fluctuations also affect the stability condition and needs to be adjusted accordingly.

4.3.4 Temporal Discretization and Stability Issues

The choice between explicit and implicit integrations can be considered as a trade off between the evolution speed of the equations or the computation time between the steps. Explicit methods, also depending on the order of the method, are generally computationally less expensive, however they require smaller steps due to instability. On the other hand, implicit integration can provide unconditional stability, however it requires the matrix solution of the whole system which could be expensive. In other words, implicit solution allows larger steps but requires more computational effort between the steps. Discretization of the individual terms in the momentum equation directly affects the efficiency of the matrix solution. For example, due to the nonlinear terms in the convective term which break the diagonal dominance, matrix solution requires iterative schemes. For this reason, usually implicit schemes such as Crank-Nicholson [47] or Adams-Moulton are considered for diffusion and explicit schemes such as Adams-Bashforth or semi-implicit methods (in the form $\mathbf{u}^n \cdot \nabla \mathbf{u}^{n+1}$) are considered for convection. Stability criteria of FV discretization is still not well studied. In [60], Fuhrmann et.al. showed that advection-diffusion of a scalar using upwind and central schemes for advection and diffusion respectively can be unconditionally stable if a fully implicit scheme is used.

Among the explicit methods, usually Runge-Kutta schemes are compared to Adams-Bashforth methods. Adams-Bashforth is a multi-step time integration method which uses previous steps in the approximation, thus it requires extra storage. Runge-Kutta is a one step method but it requires multiple evaluations at each step. Adams-Bashforth seems to be advantageous when there is no memory limitations as it requires less computational effort at each step, however at least a 3rd order scheme is necessary for accuracy, and when the order is increased the storage requirements become excessive. In fact, stability region of Runge-Kutta is considered to be larger than Adams-Bashforth. For these reasons, I used a 4th order Runge-Kutta method for the time integration:

$$\begin{aligned} k_1 &= F(t^n), \\ k_2 &= F(t^n + \tfrac{\Delta t}{2}), \\ k_3 &= F(t^n + \tfrac{\Delta t}{2}), \\ k_4 &= F(t^n + \Delta t), \\ y^{n+1} &= y^n + \tfrac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4). \end{aligned} \tag{4.45}$$

With this integration, although convection stability limit was sufficient, I had to take smaller steps for diffusion; as a result, diffusive behaviour of the convective terms became effective. I applied kriging smoothing to further improve the stability conditions. This allowed sufficient flexibility in the experiments in the current model. However in the future with the help of parallel computation a fully implicit discretization suitable for the Voronoi structure can be implemented.

4.3.5 Incompressibility

The rate of fluid flow through a unit volume or divergence of a fluid is a measure of incompressibility which is described by the continuity equation (Eqn. 4.46) .

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{V}) \tag{4.46}$$

If the fluid is incompressible divergence is zero (Eqn. 4.47) .

$$\nabla \cdot \mathbf{V} = 0 \tag{4.47}$$

Propagation of pressure waves through the fluid causes the incompressible behaviour. In other words, the inflow and outflow of fluid through the cells correspond to pressure changes.

In particle models such as Smoothed Particle Hydrodynamics, incompressibility is usually approximated by assigning an appropriate equation of state for the pressure. For instance, in [115], Monaghan defined an equation of state which depends on density as follows:

$$p \sim \rho^{\varsigma}, \tag{4.48}$$

where p and ρ indicate pressure and density respectively and ς identifies the influence of density oscillations. Incompressibility is approximated by increasing ς so that small changes in density cause high pressure oscillations. This method provides only quasi-incompressibility and does not satisfy the divergence-free condition of the vectors in the whole domain. More accurate solutions can be obtained by solving the continuity equation with the incompressibility constraint (Eqn. 4.47). Note that, this approach can also be applied to SPH [28] or other gridless methods. Numerous solution methods have been offered for the continuity equation to enforce incompressibility; these can generally be classified as artificial compressibility and fractional step (projection) methods.

4.3.5.1 Artificial Compressibility Methods

In these methods an artificial compressibility term is introduced in the continuity equation. This can be a pressure term (Eqn.4.49) as in penalty methods [149] or time derivative of pressure (Eqn.4.50) as proposed by Chorin [20] in the original artificial compressibility method.

$$\nabla \cdot \mathbf{u} + \varsigma p = 0 \tag{4.49}$$

$$\nabla \cdot \mathbf{u} + \varsigma \frac{\partial p}{\partial t} = 0 \tag{4.50}$$

The main idea is to minimize the difference in pressure or pressure derivatives by iteratively solving the momentum equation and Eqn. 4.49 or 4.50, to satisfy the incompressibility condition. At the incompressible limit time derivative in the continuity equation vanishes and density becomes constant. The artificial compressibility parameter ς is used as a convergence parameter. Note that, this idea is similar to the modified density solution of particle models, however it is repetitive and provides higher order accuracy. Originally the method was developed for steady state problems or for unsteady flows where only the asymptotic steady state is required (since the artificial time derivative of pressure vanished in the steady state limit). However later Kim and Moin [95] developed an implicit solution which allowed larger time steps, in this case the method does not need the iterations in between the time steps and it could be applied to unsteady flows. The main drawback of this method is its poor convergence ability, also it is costly as it requires the solution of the momentum equation at each time step.

4.3.5.2 Fractional Step Methods

The fractional step methods, also called projection methods, which are pioneered by Chorin [21], make use of the Hodge decomposition theorem. Hodge decomposition states that a vector can be uniquely decomposed into a divergence free vector and a gradient of a scalar. When the momentum equation is arranged to decompose velocity in this way,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla p = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \nu \nabla^2 \mathbf{u}, \tag{4.51}$$

$$\mathbf{u} = \mathbf{u}_{\text{div}} + \Delta t \nabla \phi, \tag{4.52}$$

the initial velocity obtained from momentum equation can be projected onto the divergence free vectors as follows:

$$\mathbf{u}_{\text{div}} = \mathcal{P}(\mathbf{u}), \tag{4.53}$$

$$\nabla \phi = (I - \mathcal{P})(\mathbf{u}), \tag{4.54}$$

where $\mathbf{u}_{div}$ represents the divergence free velocity. Taking the divergence of Eqn. 4.54 to obtain ϕ yields the Poisson equation:

$$\nabla^2\phi = \frac{1}{\Delta t}\nabla\cdot\mathbf{u}. \tag{4.55}$$

To sum up, by using the Hodge decomposition property, first an intermediate velocity is obtained from the momentum equation, then this vector is projected onto the space of divergence free fields to obtain the velocity at the next time step. To project the vectors, first Poisson equation is solved to obtain ϕ, then velocity is updated to include the pressure gradient.

Several approaches have been proposed for the solution of this system using different variations of the original idea. A general formulation can be given by:

$$\mathbf{u}^* = \mathbf{u}^n - \Delta t(\mathbf{u}^n\cdot\nabla)\mathbf{u}^n - \Delta t\beta\nabla p^n + \Delta t\nu\nabla^2\mathbf{u}^n, \tag{4.56}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t\nabla\phi, \tag{4.57}$$

$$p^{n+1} = \beta p^n + \phi. \tag{4.58}$$

In general, the solutions can be classified as iterative and one-step methods. In one-step solutions, the Equations 4.56, 4.57, 4.58 are solved once at each time step. In the basic projection method as first proposed by Chorin [21] β is zero, so the pressure gradient is dropped from the momentum equations and ϕ is initialized to the previous step. This system is solved in one step. Since neglecting the pressure gradient in the momentum equation can decrease accuracy, for this reason, [6] first proposed retaining the pressure term in the momentum, (that is, $\beta \neq 0$) for a second order accurate projection. In this case best available estimate is used for pressure and ϕ is initialized to zero at the beginning of each step. This method is also called the pressure correction method.

In the iterative solution, these equations are repeated until the divergence in the whole domain becomes less than a predefined value. Note that, here the number of steps refers to the number of times the momentum and pressure corrections are iterated. Otherwise even in one step methods, since the Poisson solution usually requires iterations, using iterative methods in the incompressibility solution is almost inevitable. One-step solution might seem less expensive at first sight, however in this case Poisson solution must be very accurate to satisfy the divergence free condition in one step. Iterative methods, on the other hand, require more computational effort both for the iterations and the solution of momentum equation at each step.

A different variation of the iterative pressure correction method is the SIMPLE (Semi-Implicit Method for Pressure Linked Equations) scheme [47]. In this method velocity and pressure are adjusted by a correction parameter to obtain new values which satisfy incompressibility:

$$\mathbf{u} = \mathbf{u}^* + \mathbf{u}', \tag{4.59}$$

$$p = p^* + p'. \tag{4.60}$$

The relation between the velocity and pressure corrections is obtained by substituting the corrected values into the momentum equation and by using an approximation based on neglecting the gradient terms which include the velocity corrections from the neighbour cells. This is the main simplification used in the SIMPLE method; since the system is solved iteratively this is considered a safe assumption. As a result, the velocity correction is related to the pressure correction as follows:

$$u' = \frac{1}{A_x}\frac{\partial p'}{\partial x}, \tag{4.61}$$

where $\frac{1}{A_x}$ represents an area related term. Next by substituting the corrected velocity into the continuity equation a Poisson like equation is derived from which the pressure correction terms are obtained. Again this equation is derived by using a similar assumption by removing the nonorthogonal terms and it is solved iteratively. This is a commonly used method on unstructured grids due to non-orthogonal simplifications. It was originally developed for steady flows and lately it has been adapted to transient flows [3].

4.3.5.3 Voronoi Based Finite Volume Projection

Convergence, smoothness and stability of the incompressibility solution usually depend on the Poisson solution and the characteristics of spatial discretization. I have formulated a finite volume projection method based on the marbling Voronoi grid structure, for the solution of continuity equation, where β in Eqn. 4.56 is defined as 1 and ϕ is initialized to zero at the beginning of each step. As a result of experimenting with different β values and I obtained the best convergence results with $\beta = 1$ value.

Poisson equation can be solved by explicit, semi-implicit iterative or matrix decomposition methods. The main issue in iterative schemes is convergence. If the system is not symmetric (and lacks diagonal dominance) usually iterative schemes can't achieve convergence. Some of the common iterative methods are Point Jacobi and Successive Over-relaxation (SOR) schemes such as Gauss-Seidel method. Point-Jacobi uses an explicit iteration formula and it has poor convergence conditions($O(N^2)$). In SOR methods grid is updated successively, that is, new values are used as the grid is scanned; updating orders are designed to improve the convergence rate (which is usually considered $O(N^{3/2})$). Matrix solutions can be fast and stable depending on the characteristics of the matrix which are determined by discretization and mesh properties. However they are usually computationally demanding.

Since the grid structure is mainly symmetric, I was able to use an iterative method for Poisson equation. Suppose the following iterations are defined to satisfy the divergence-free condition:

$$\phi^{k+1} = \phi^k - \bar{\epsilon}\nabla\cdot\mathbf{u}^{n+1}. \tag{4.62}$$

Poisson solution (Eqn. 4.63) can be obtained, by substituting the velocity corrections (Eqn.4.57) into Eqn. 4.62:

$$\phi^{k+1} = \phi^k - \epsilon\nabla\cdot\mathbf{u}^* + \Delta t\bar{\epsilon}\nabla\cdot(\nabla\phi^k), \tag{4.63}$$

where k indicates the iteration number and $\bar{\epsilon}$ represents a convergence parameter.

I used a Dufort-Frankel scheme, which is considered a Successive Over-relaxation method. This method was applied by Chorin in [21], for Poisson solution on a rectangular grid. In Dufort-Frankel method [47] only ϕ value at the current position is represented semi-implicitly as an average of current and next step values,

$$\phi_{i,j} = \frac{1}{2}(\phi_{i,j}^{k} + \phi_{i,j}^{k+1}), \tag{4.64}$$

and the rest of the terms in Eqn.4.63 are represented explicitly. This solution is obtained by enforcing Young's matrix condition to provide optimum relaxation.

In discretizing the projection system, same gradient and divergence operators are used in momentum and Poisson equations and velocity and pressure corrections, as described in Section 4.3, to achieve consistency. In order to formulate the Poisson discretization, I first substitute Eqn. 4.64 into the Poisson Eqn.4.63. From this an extra term emerges,

$$\frac{1}{2}\boldsymbol{\nabla}\cdot(\boldsymbol{\nabla}\phi^{k+1}), \tag{4.65}$$

which is discretized as:

$$\begin{gathered}\tfrac{1}{2}\textstyle\sum_j(\boldsymbol{\nabla}\phi_{\mathbf{ij}}^{\mathbf{k+1}}\cdot\mathbf{n})L_{ij} = \tfrac{1}{2}\textstyle\sum_j \tfrac{1}{2}(\boldsymbol{\nabla}\phi_{\mathbf{i}}^{\mathbf{k+1}} + \boldsymbol{\nabla}\phi_{\mathbf{j}}^{\mathbf{k+1}})\cdot\mathbf{n}_{\mathbf{ij}}L_{ij} = \\ \textstyle\sum_j \tfrac{1}{2}[(\displaystyle\sum_m(\phi_m^k - \phi_i^{k+1})\mathbf{n}_{mi}L_{mi}) + (\sum_m(\phi_m^{k+1} - \phi_j^k)\mathbf{n}_{mj}L_{mj})]\cdot\mathbf{n}_{ij}L_{ij},\end{gathered} \tag{4.66}$$

where index $k+1$ is used in summations which included ϕ_i. Arranging this equation to leave out the ϕ_i^{k+1} terms gives us:

$$\frac{1}{2}\boldsymbol{\nabla}\cdot(\boldsymbol{\nabla}\phi^{k}) + \bar{\theta}\frac{1}{2}\boldsymbol{\nabla}\cdot(\boldsymbol{\nabla}\phi^{k+1}), \tag{4.67}$$

where $\bar{\theta}$ represents the coefficients,

$$\bar{\theta}_i = \sum_j \left(\sum_m \mathbf{n}_{mj}\, L_{mj}\right)\cdot\mathbf{n}_{ij}L_{ij}. \tag{4.68}$$

This also shows that without the orthogonality property, obtaining the coefficients would be very difficult. Note that in Equations 4.66 and 4.67 in gradient approximation current point values are taken as $\phi_i/2$ and $L_{ij}/2$.

Then substituting these into the Poisson equation (Eqn. 4.63) gives the following Dufort-Frankel iteration scheme for the solution of the Poisson equation:

$$\phi_i^{k+1} = \frac{\phi_i^k - \bar{\epsilon}\nabla\bullet\mathbf{u}^{*} + \Delta t\bar{\epsilon}\boldsymbol{\nabla}\cdot(\boldsymbol{\nabla}\phi_i^k)}{1 + \Delta t\,\bar{\epsilon}\,\bar{\theta}_i}. \tag{4.69}$$

With this method I was able to achieve convergence between 5 to 7 iterations. The parameters $\bar{\epsilon}$, and Δt should be adjusted to obtain the optimum convergence rate. I compared the results with the Point-Jacobi method; Dufort-Frankel provides faster convergence and it is unconditionally stable while Point-Jacobi is restricted by the condition $\frac{\Delta t}{\Delta x} < \frac{1}{4}$. Note that, given the properties of the Voronoi discretization BiConjugateGradient matrix solution could also be an alternative in this model. BiConjugateGradient [47] is an improvement on Conjugate-Gradient method (which requires symmetry) to support asymmetric conditions. However to my knowledge it has the same convergence rate as SOR methods. I also compared the results with the artificial compressibility method; although they produce similar results projection solution provides faster convergence (with higher accuracy in divergence-free condition).

One problem with this method is that it usually does not provide divergence free vectors after one step due to the first order approximation of the Poisson solution. Hence, in order to improve accuracy, I built an iterative system where velocity and pressure correction steps (Equations 4.57,4.58) are repeated until divergence free conditions are satisfied in the whole grid below a threshold value. These iterations are performed for each momentum step, that is momentum equation is not solved in each iteration for time efficiency.

Another problem with the continuity solution on a non-staggered grid is the checkerboard effect arising from the finite-volume discretization; this velocity-pressure decoupling occurs when the velocity which is obtained from the momentum equation becomes unlinked from the pressure values which are obtained from the approximated cell face velocities. As a result, pressure oscillations might occur. This problem is commonly solved used using Rhie-Chow interpolation [126]. In this approach, a pressure gradient term is included in the cell face velocity approximation in convective flux to couple velocity and pressure. Derivation of this interpolation idea is based on the SIMPLE algorithm [47]. Recently Date [30] showed that the checkerboard problem can also be solved by smoothing the pressure correction without the need for a modification in the momentum discretization. I solved this problem by smoothing the ϕ values with a kriging technique (see next section).

Finally, the incompressibility algorithm in general form can be given as follows:

```
for each time step do
  Solve momentum equation to obtain $u^*$
  for each iteration do
    Initialize $\phi$
    Solve Poisson equation to obtain $\phi$
    Smooth $\phi$
    Update pressure correction (Eqn. 4.57)
    Update velocity correction (Eqn. 4.58)
  end.
end.
```

4.3.6 Boundary Conditions

Two types of boundaries are defined: fixed and moving boundaries. Fixed boundaries of the fluid grid represent the walls of the marbling tray. Marbling stylus or comb is defined as a moving boundary. Both are defined with *no-slip* conditions which means that there is no motion relative to the solid surface of the boundary. *No-slip* condition also states that the fluid particles cannot penetrate the solid boundaries.

In order to satisfy the *no-slip* condition on the fixed boundaries, normal and tangential velocities on the walls are set to zero,

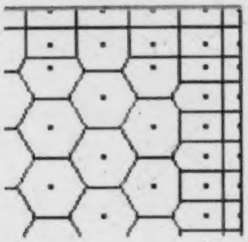


Figure 4.5: Boundary configuration near the border.

$$(u, v) = 0,\ \Omega \tag{4.70}$$

where, Ω represents boundary grid points. Boundary conditions for the pressure projection method requires more effort and can be problematic. Recall that, in the projection method Poisson equation is solved to obtain a pressure correction term, and pressure gradient is included in the momentum equation. This formulation can affect the accuracy of the pressure boundary condition; since the pressure correction term, ϕ, is included in the momentum equation, the boundary condition imposed on ϕ must be designed to satisfy the momentum conservation. Considering this, the following second order conditions have been defined [6]:

$$\nabla\phi\cdot\mathbf{n} = (\mathbf{u}^* - \mathbf{u})\cdot\mathbf{n},\ \Omega. \tag{4.71}$$

Since the velocity on the no-slip boundary is fixed (i.e. not updated) the intermediate velocity $\mathbf{u}^*$ is assumed to be constant; Eqn. 4.71 reduces to:

$$\nabla\phi\cdot\mathbf{n} = 0,\ \Omega. \tag{4.72}$$

In order to simplify the discretization of this equation on the Voronoi structure, I defined a parallel set of grid points in contact with the walls. In this way, each boundary point is forced to have only one contact with the interior which is normal to the boundary (Figure 4.5). As a result, along the walls, discrete form of the boundary condition (Eqn.4.72) can be represented by:

$$\phi_i = \phi_\Omega, \tag{4.73}$$

where i indicates the interior point that is adjacent to the boundary cell.

Modeling the moving boundaries accurately is a difficult issue. Moving the grid along the boundary is one option. However this is a tedious and inefficient computation for the marbling simulation. Another approach is embedding the boundary in the fluid grid. In the marbling model, external moving boundary is introduced in the flow as a new point immersed in the fluid grid. Multiple external boundaries can be introduced in the flow simultaneously: they can either move independently, or can be used to define the faces of an object or a line. However in case of rigid boundaries that exert force on the flow (i.e. boundaries that are not affected by the flow) as in marbling, only the boundary surface affects the flow. Thus, there is not much point in defining solid objects, in terms of creating different flow behaviours, instead the user can define different geometric configurations for the surfaces. The position and velocity direction of the boundary points are defined by the user. In the experiments velocity is usually defined in the tangential direction to the boundary path. The boundary velocity is fixed, that is, not updated in the equations. And in case of point boundaries, I simply assigned zero pressure correction on the boundary point.

The effect of moving surfaces on the fluid at high Reynolds numbers is another issue. Small perturbations become important in this case, and discretization usually requires adaptive grid refinement along the surface. Projection step boundary conditions for the moving surface is given by:

$$\nabla\phi\cdot\mathbf{n} = \mathbf{u}_{\Omega}\cdot\mathbf{n},\ \Omega. \tag{4.74}$$

Defining the velocity normal to the boundary surface simplifies the discretization somewhat. However, still, imposing this condition is not as straightforward as in the case of fixed boundaries. Basically boundary value needs to be extracted from the above equation to be represented in terms of its neighbours. However also due to the irregular geometric configurations that occur as the surface moves, the resulting approximate representation affects the convergence rate of the Poisson solution and it arouses instabilities. In any case, this level of accuracy in the boundary representation would not be sufficient to model high Reynolds flows. Simulation of boundary effects at higher Reynolds numbers is considered under a different approach as discussed in the next chapter. Here, I used an approximate representation and assigned zero pressure on the boundary points, this level of accuracy was sufficient for viscous flow modeling.

During its motion, if a boundary point becomes closer to a point less than a threshold value, then the point is not inserted and the closest grid point is treated as a boundary. In other words, depending on the grid size (average Voronoi cell area), and a given threshold value, the boundary points can either be inserted or immersed in the fluid grid. External points are not allowed to enter the Voronoi region of the borders in order not to change the Voronoi structure there. I used Bowyer-Watson algorithm[10] for the insertion of points in the Voronoi diagram (Figure 4.6). Bowyer-Watson algorithm provides fast local insertion of points in a Voronoi diagram. It is basically given as follows. First locate the triangle that contains the point to be inserted. Next, find all the neighbouring triangles (that is, the triangles which contain any of the vertices of the triangle which contains the new point), whose circumcircle contains the new point. Then eliminate the interior connections of each such triangle to form a convex hull. Convex hull formation in this way is guarantied by the properties of the Delaunay triangulation (see section 4.2). Finally connect the new point to the vertices of this convex polygon.

In case boundary is inserted (rather than immersed), the threshold value is adjusted high enough not to let the formation of very thin irregular triangles that might cause instabilities. Insertion could be useful when the average cell area is big and the insertion of a new point makes a visible difference, or just when high resolution boundary computation is required. However I haven't found a use for this effect in the experiments.

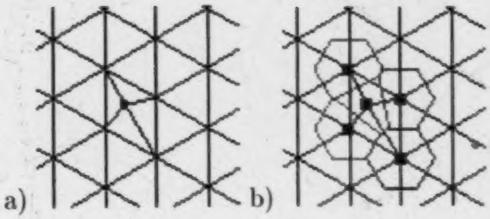


Figure 4.6: Neighbourhood connections and Voronoi regions are updated when a new particle is inserted.

4.4 Interpolation and Smoothing of Fluid Vectors

Interpolation of velocity vectors is needed for the solution of advection equation on the fine grid. In addition a smoothing function is required to filter the velocity fluctuations in turbulence modeling. The following criteria are considered for smoothing/averaging the vectors on the fluid grid and for the interpolation on the image grid.

- Accuracy and stability: Interpolation error should be minimized and the method should be able to capture the characteristics of the data accurately, that is, spatial vector evolution on the fine grid should be consistent with the vectors on the fluid grid. As a result of instabilities in the interpolation oscillations might occur even though the data fits the given values at the grid points. This is also important when this interpolation is used in the calculation of the momentum fluxes.

- Continuity: Smoothness of the approximation is important for the stability, and continuity of the advection results. Continuity affects the advection results especially in high gradient regions.

- Computational efficiency (computational complexity and memory requirements): Since interpolation needs to be performed at each fluid step computational demands should be minimized. Memory requirements is of secondary importance. Both for computational efficiency and stability, local transformation should be considered; as the grid resolution increases global transformation becomes unfeasible, besides when the number of data values are increased the interpolation becomes more sensitive to small changes, hence more susceptible to oscillations.

4.4.1 Scattered Data Interpolation Techniques

Scattered data fitting methods are considered for the interpolation on the Voronoi structure. These are categorized as least squares methods and splines.

4.4.1.1 Moving Least Squares

Approximation using moving least squares (MLS) has already been introduced in Section 4.1. The same system of equations can be designed to interpolate the data with additional constraints. In this section, well-known Shepard's interpolants will be discussed. Shepard's [132] interpolation based on the weighted averaging of the data is considered the first order MLS representation, and it is given as follows:

$$F(\mathbf{x}) = \sum_{i=1}^{N} w_i(\mathbf{x}) f_i, \tag{4.75}$$

$$w_i(\mathbf{x}) = \frac{d_i^{-k}}{\sum_{i=1}^{N} d_i^{-k}}, \tag{4.76}$$

where F represents the continuous approximation, and f_i indicates the data values at the given points. In addition, $d_i = \|\mathbf{x} - \mathbf{x}_i\|$ denotes the distance from the current point $\mathbf{x} \in \mathbb{R}^{\kappa}$ to the data point $\mathbf{x}_i$, thus weights, w_i, depend on the distance from the data points. Usually k is given as 2 ($k > 1$ assures the continuity of the derivatives). In its original form Shepard's method is a global transformation, and it causes all the points away from the window center to move towards the average of all $\mathbf{x}_i$, in other words all the points tend to be warped towards the center of the image. In Modified Shepard method [55], this transformation is localized to provide higher order accuracy. The weight functions in this case are defined as:

$$w_i(\mathbf{x}) = \frac{\bar{w}_i^{-k}}{\sum_{i=1}^{N} \bar{w}_i^{-k}}, \tag{4.77}$$

$$\bar{w}_i = \begin{cases} (\frac{r_c - d_i}{Rd_i}) & d_i < r_c \\ 0 & d_i \geq r_c \end{cases}, \tag{4.78}$$

where r_c indicates the radius of influence. In Eqn.4.75 f_i are replaced by a quadratic function which equals the data at the grid points ($q_i(\mathbf{x}_i) = f_i$). In order to determine the local interpolant, estimation of derivatives at data points are required. The coefficients of the interpolant ($q_i(\mathbf{x})$) are obtained by minimizing the following error function:

$$\varepsilon_i = \sum_j \bar{w}_j(\mathbf{x}_i)[q_i(\mathbf{x}_j) - \mathbf{x}_j]^2. \tag{4.79}$$

This is a least squares problem which is usually solved by solving the set of equations that are obtained from the derivatives of ε_i according to the coefficients.

4.4.1.2 Splines (Radial Basis Functions)

Another approach is to construct the interpolant as a linear combination of radial basis functions:

$$F(\mathbf{x}) = \sum_{i=1}^{N} w_i(\|\mathbf{x} - \mathbf{x}_i\|)\lambda_i + \sum_{j=1}^{m} b_j q_j(\mathbf{x}). \tag{4.80}$$

Here $w_i(r)$ represents the radial basis functions which can be defined as one of the several choices such as Thin Plate Splines, Gaussian or multiquadric functions etc.:

$$\begin{array}{rl} \text{Thin Plate Spline,} & w(r) = r^2 log(r) \\ \text{Gaussian,} & w(r) = e^{-cr^2} \\ \text{Multiquadric,} & w(r) = \sqrt{r^2 + c^2} \end{array} \quad . \tag{4.81}$$

In Eqn.4.80 $q(\mathbf{x})$ represents a polynomial and it is an optional parameter. The coefficients λ_i are obtained by enforcing $F(\mathbf{x})$ to satisfy the conditions:

$$F(\mathbf{x}_i) = f_i, \tag{4.82}$$

$$\sum b_i q(\mathbf{x}_i) = 0. \tag{4.83}$$

The second condition is imposed to complete the set of equations that is required to solve the system when polynomials are included. This leads to the following system of matrices:

$$\hat{\mathbf{W}}\hat{\boldsymbol{\lambda}} = \mathbf{F}, \tag{4.84}$$

$$\begin{vmatrix} \mathbf{W} & \mathbf{Q} \\ \mathbf{Q}^T & 0 \end{vmatrix} \times \begin{vmatrix} \lambda \\ b \end{vmatrix} = \begin{vmatrix} \mathbf{F} \\ 0 \end{vmatrix}, \tag{4.85}$$

$$\hat{\boldsymbol{\lambda}} = \hat{\mathbf{W}}^{-1}\mathbf{F}, \tag{4.86}$$

where $W_{ij} = w(\|\mathbf{x}_i - \mathbf{x}_j\|)$ and $Q_{ij} = q_j(\mathbf{x}_i)$. This system is designed to minimize the error function:

$$\varepsilon = \sum_{i=1}^{N} [f_i - F(\mathbf{x}_i)]^2. \tag{4.87}$$

The first RBF method, as proposed by [71], used multiquadrics as the basis function and did not include the polynomials. However, later it was observed that without the polynomials, multiquadrics could produce distortions due to the nonlinearity of the basis functions. Without the polynomials the coefficient matrix is positive definite and symmetric, hence matrix inversion can be computed by Cholesky decomposition and back substitution. However when the polynomials are included, even though the matrix is still symmetric it is not positive definite anymore, as a result, the linear system requires more expensive solutions such as diagonal pivoting [14] which is an iterative method.

Smoothing of the data with RBF can be obtained by adding a smoothing term in the error minimization; the coefficients are then obtained from,

$$\hat{\boldsymbol{\lambda}} = (\hat{\mathbf{W}} + \iota\mathbf{I})^{-1}\mathbf{F}, \tag{4.88}$$

where ι indicates the smoothness parameter.

Splines provide the smoothest results among other interpolation methods, they are flexible in the choice of the basis functions, and by using the weight functions the shape and continuity of the interpolation can be controlled. In addition, coefficient matrix is invertible, that is, it has a unique solution. In its original form RBF is designed as a global transformation, however it can be localized to an extent by limiting the range of the basis functions. Although the range of influence and so the data approximation are defined locally, the solution system for the coefficients stays compact for the whole data set and still requires the inversion of a large matrix which can be ill-conditioning. For the solution of this problem, some methods which exploit the sparsity of the matrix have been proposed. For example Beatson *et.al.* developed a multistep method in [5] for the fast evaluation of the matrix system. This approach suggests an iterative solution of the overlapping subspaces in the matrix which are based on the scales of RBF. However these are still computationally limited and decrease the flexibility of the interpolation. Another disadvantage is that the splines tend to produce oscillations; in global solution sensitivity to small changes increases. Solving the coefficients in local overlapping regions around each point, and then averaging the coefficients that are computed more than once could also be a solution to the localization problem. However this idea requires further analysis regarding the accuracy of the interpolation.

I should also mention the interpolation methods which use triangulation based basis functions. In particular, Natural Neighbour Interpolation [146] would be suitable to my model since it is based on the Delaunay triangulation of the data points. This method probably can produce as smooth and accurate results as Splines with local computation which although may not be very efficient. However it does not provide smoothing of the data.

4.4.1.3 Kriging

Kriging is an approximation method which uses weighted linear combinations for the estimation of data points. Although it is very popular in geostatic studies it has been mostly overlooked in computer graphics applications. This method is based on the assumption that the underlying spatial process is a continuous random field which is described by a spatial covariance function. In kriging the estimate of a point using N data points is given by:

$$F^*(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i f_i. \tag{4.89}$$

The weights λ_i are chosen to minimize the error variance:

$$\begin{aligned} \sigma_e^2 &= E[(F(\mathbf{x}) - F^*(\mathbf{x}))^2] = E[F(\mathbf{x})^2] + E[F^*(\mathbf{x})^2] - 2E[F(\mathbf{x})F^*(\mathbf{x})] \\ &= \mathrm{var}(F(\mathbf{x})) + \sum_{i=1}^{N} \lambda_i \mathrm{cov}(\|\mathbf{x}_i - \mathbf{x}\|) - 2\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j \mathrm{cov}(\|\mathbf{x}_i - \mathbf{x}_j\|) \end{aligned} \tag{4.90}$$

where $\mathrm{var}(x)$ and $\mathrm{cov}(x)$ represent the variance and covariance functions respectively. In kriging, data is composed of a mean and an error term:

$$F(\mathbf{x}) = m + e. \tag{4.91}$$

There are various types of kriging based on these principles. In ***simple kriging*** the mean is constant (indicating the stationarity of the random process). ***Ordinary kriging*** [27] assumes that the data value consists of a constant (but locally unknown) mean m and a constant error term e. Thus in ordinary kriging error mean is assumed to be zero, therefore estimation is expected to be unbiased. Zero error mean yields the condition in Eqn. 4.93 as follows:

$$E\{F^{*}(\mathbf{x}) - F(\mathbf{x})\} = \sum_{i}^{N} \lambda_i m(\mathbf{x}) - m(\mathbf{x}) = 0, \tag{4.92}$$

$$\sum_{i=1}^{N} \lambda_i = 1. \tag{4.93}$$

This condition forces the data to pass through the data points. In other words, interpolation accuracy is increased by enforcing zero error mean. In *universal kriging* [27], mean m is not constant but it depends on the position; a polynomial is added to the estimation function as in RBF, which accounts for the trend in the data. It is useful when the data point variation is irregular and the stationarity assumption is not valid. Since the point distribution is quite uniform on the Voronoi fluid grid I did not find it necessary to define a trend and used ordinary kriging in the marbling model.

In order to obtain the weights in ordinary kriging, the unbiased condition requires the addition of a Lagrange multiplier term, ℓ, in the error minimization:

$$\hat{\sigma}_e^2 = \sigma_e^2 - 2\ell\left(\sum_{i=1}^{N} \lambda_i - 1\right). \tag{4.94}$$

The weight parameters λ_i are obtained by setting the partial derivatives of $\hat{\sigma}_e^2$ with respect to λ_i and the Lagrange parameter ℓ to zero, which then leads to the system of equations given below (Equations 4.95,4.96,4.97). In other words the weights are designed to minimize the estimation variance.

$$\mathbf{C}\boldsymbol{\lambda} = \mathbf{C}^{*} \tag{4.95}$$

$$\begin{vmatrix} \mathrm{cov}(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \mathrm{cov}(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \cdots & \mathrm{cov}(\|\mathbf{x}_1 - \mathbf{x}_n\|) & 1 \\ \mathrm{cov}(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \mathrm{cov}(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \cdots & \mathrm{cov}(\|\mathbf{x}_2 - \mathbf{x}_n\|) & 1 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \mathrm{cov}(\|\mathbf{x}_n - \mathbf{x}_1\|) & \mathrm{cov}(\|\mathbf{x}_n - \mathbf{x}_2\|) & \cdots & \mathrm{cov}(\|\mathbf{x}_n - \mathbf{x}_n\|) & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{vmatrix} \times \begin{vmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \\ \ell \end{vmatrix} = \begin{vmatrix} \mathrm{cov}(\|\mathbf{x}_1 - \mathbf{x}_0\|) \\ \mathrm{cov}(\|\mathbf{x}_2 - \mathbf{x}_0\|) \\ \vdots \\ \mathrm{cov}(\|\mathbf{x}_n - \mathbf{x}_0\|) \\ 1 \end{vmatrix} \tag{4.96}$$

$$\boldsymbol{\lambda} = \mathbf{C}^{-1}\mathbf{C}^{*} \tag{4.97}$$

The similarity to RBF is evident from this system. In fact, RBF splines is considered a special case of kriging. [74] showed that RBF is equivalent to *dual kriging*. (See also [159] for a more general and theoretical comparison.) The weights are determined according to the covariance function which depends on the distance between the observations: the observations closer to the prediction point get more weight. Among the several choices for the covariance are spherical, logarithmic, Gaussian and Bessel functions which are defined as radial functions as in RBF:

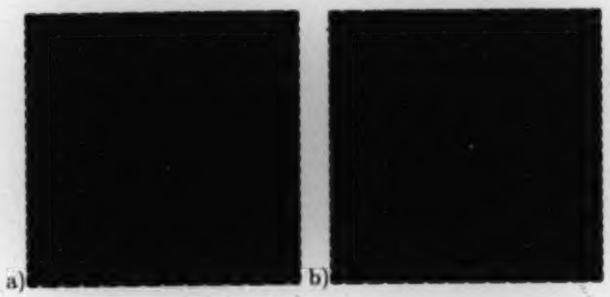


Figure 4.7: Interpolation range for two different positions of the estimation point.

$$\text{Spherical} : \text{cov}(r) = \begin{cases} (1 - 1.5\frac{r}{r_c} + 0.5(\frac{r}{r_c})^3) & 0 < r \leq r_c \\ 0 & r \geq r_c \end{cases}, \tag{4.98}$$
$$\text{Logarithmic} : \text{cov}(r) = 1 - log(r + r_c)$$

where r and r_c indicate distance and range values respectively.

4.4.2 Kriging in the Marbling Model

Kriging requires the weights to be computed for each estimation point, rather than for each data point as RBF. This brings a major computational difference considering the resolution difference between the fluid and image grids. On the other hand, together with the covariance properties it also brings the flexibility of localizing the estimation process. Originally kriging is defined as a global approximation. I localized the process by limiting the range of influence; at each estimation point only the data points within a given radius of this point are included in the estimation (Figure 4.7). Since the effect of data points decrease as the points get further from the estimation point, this is a reasonable assumption. However, this localization might cause discontinuities in the weights at the intersections of the local regions if a small radius is used. I found that, a range which includes the second level neighbours of the closest data point was sufficient for continuous transition in the overlapping regions. Furthermore this localization reduced the oscillations. I also found that oscillatory behaviour as well as the interpolation smoothness can be controlled by adjusting the covariance function. Let us define a general covariance function based on an exponential form with parameters that allow the control of these characteristics:

$$\text{cov}(r) = \begin{cases} \exp(-s[\frac{r}{r_c}]^{C^n}) & r > 0 \\ \hat{\sigma}_\eta^2 + 1.0 & r = 0 \end{cases}, \tag{4.99}$$

where r denotes the distance between the points, and r_c indicates the range.

Characteristics of the covariance function that affect the interpolation are continuity of the slope in the origin and continuity and position of transition to zero. In this function, s and r_c control the range of influence, and C^n defines the continuity of the initial slope. Figure 4.8a shows the covariance properties for varying s values with $C^n = 2$ and Figure 4.8b,c,d,e shows the corresponding interpolation graphs for the given data. (In all the figures $r_c = 0.34641$ on a 2×2 grid.) A local radius is defined in the algorithm to determine the maximum number of data points included in the estimation; it is defined as a range within a second level neighborhood of an average cell in the bulk of the fluid in these experiments. If the local radius which determines the number of points is longer than the range, r_c, defined in the covariance function oscillations occur, if it is much less than this range than a sharp cut-off takes place in the interpolation shape at the data points. I assigned r_c to the local radius value and used s to adjust the smoothness of the transition. The order of the exponential, C^n determines the continuity of the interpolation in between the data points. If the order is not high enough the interpolation becomes noisy. However when the order is increased the range should be adjusted as well to prevent the oscillations since the shape of the function at transition to zero will also change (Figure 4.9).

For smoothing the data, ***nugget effect*** is used. In this case, error component of the data, e, is assumed to have a nonzero variance, $\hat{\sigma}_\eta^2$. As a result of the minimization (Eqn. 4.90), the following solution matrix is obtained which is similar to splines,

$$\boldsymbol{\lambda} = (\mathbf{C} + \hat{\sigma}_\eta^2 \mathbf{I})^{-1} \mathbf{C}^*. \tag{4.100}$$

Nugget value, $\hat{\sigma}_\eta^2$, arises from the influence of the error variance in the minimization. It has a smoothing effect on the approximation. If the nugget is zero, kriging is an exact interpolation, otherwise the predictions are smoothed. In a way, increasing the nugget value corresponds to easing the minimization process by allowing an error value. This also explains the decrease in the computation time as nugget increases. Evidently from Eqn. 4.100, nugget can be defined as a discontinuity in the covariance function. The jump in the covariance represents the variation of the predictions from the exact values at data points.

Figure 4.10 shows the smoothing with two different nugget levels for a given data set. The vector interpolation is obtained by using the same weights for each component of the vector. Smoothing of two dimensional data is demonstrated in Chapter 5.

To sum up, kriging provides smoothing and interpolation of the data with the same set of equations. Robustness and smoothness of the approximation and the flexibility in its implementation are some of the factors that make it suitable for the marbling model.

I should also mention that the estimation error in kriging can be further decreased by using robust kriging algorithms. These methods such as Bayesian kriging [27] have been developed to remove the error by iteratively applying the minimization based on a trend analysis, however they are too costly for an interactive simulation.

The covariance matrix is symmetric but indefinite due to the Lagrangian multiplier, thus it requires singular value decomposition. I used CLAPACK's diagonal pivoting function for the computation of the linear system. Although localization reduces the computation time considerably, matrix inversion for each image point still constitutes an expensive task. In fact, calculation of the weights on a 500×500 grid with each point having at least 20 estimation points took approximately 18 minutes on a Xeon 1.6 GHz. PC. For this reason, the weights corresponding to each point on the image grid, for each nugget value, are precalculated and restored in the beginning of the simulation. Calculation of weights during the simulation might be necessary only when external points are introduced in the fluid grid and Delaunay structure is updated. In this case, only the weights for the

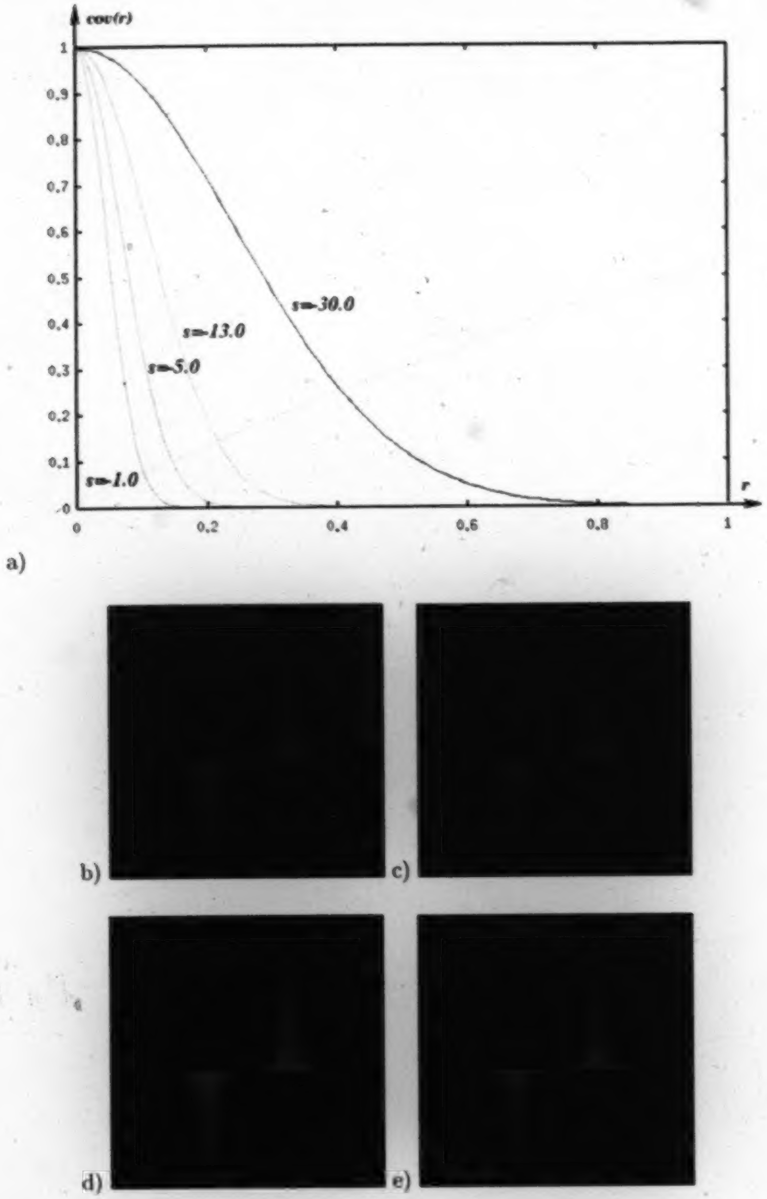


Figure 4.8: Covariance function (with $\hat{\sigma}_\eta^2 = 0$) for varying s values: a) the corresponding interpolated data b) $s = -1.0$, c) $s = -5.0$, d) $s = -13.0$, e) $s = -30.0$.

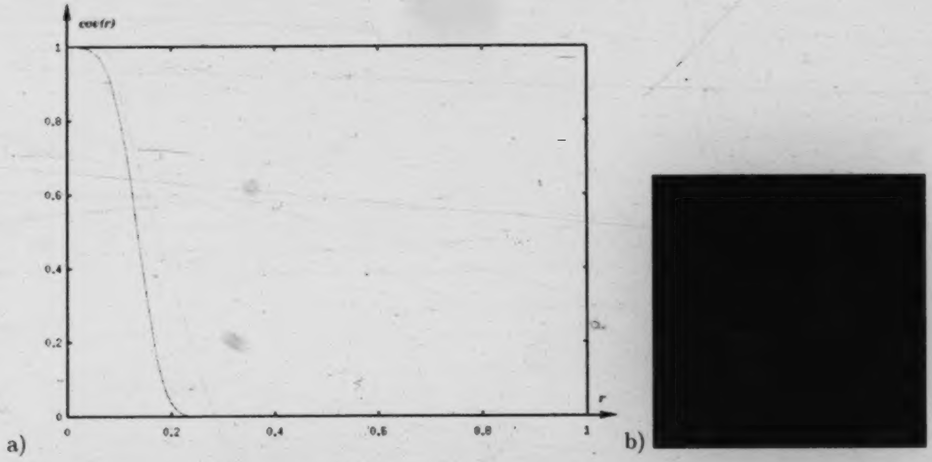


Figure 4.9: Interpolation: b) for the covariance function: a) with $s = -30, C^n = 4$

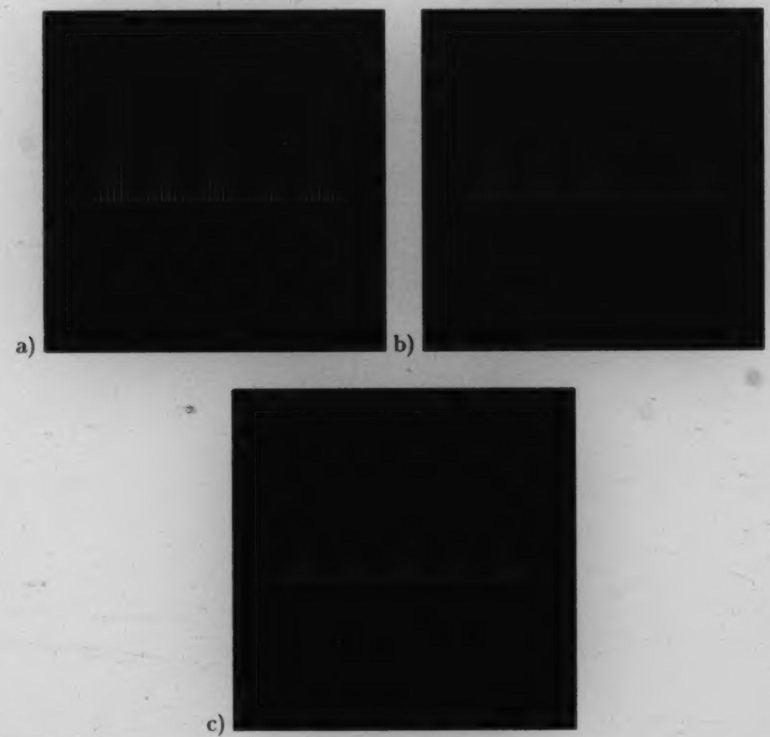


Figure 4.10: Interpolation with nugget levels: a)$\hat{\sigma}_\eta^2 = 0.0$, b) $\hat{\sigma}_\eta^2 = 0.5$, c) $\hat{\sigma}_\eta^2 = 1.0$

points which are affected by the new neighborhood connections are updated. In fact, this situation is not so likely to happen if the grid resolution is high. Besides in some cases, for time efficiency, I used the weights from the original grid although the fluid equations were solved according to the updated connections; this did not affect the results when the details smaller than the cell size were irrelevant. Also the data points were prevented from being too close in the algorithm, in order not to trouble the matrix inversion.

4.4.3 Summary

The use of kriging in the marbling model can be summarized as follows:

1. It interpolates the velocity vectors in between the grid points to obtain the fine grid velocities.
2. It smooths/averages the mesoscale fluctuations in turbulence modeling.
3. It stabilizes the finite volume discretization.
4. It removes the velocity-pressure decoupling arising from the Poisson solution on the non-staggered grid.

4.5 Transport Model

The material transport is performed on the fine (image) grid which is defined as a two-dimensional uniform rectangular grid. Advection of a scalar is described by the equation:

$$\frac{\partial C}{\partial t} = -\mathbf{U}\cdot\nabla C, \tag{4.101}$$

where C is a scalar quantity which might represent the density of the pigments or the values in each color channel, and $\mathbf{U}$ represents the interpolated velocity. Contrary to its simple representation, numerical solution of this equation has been one of the most challenging problems in computational fluid dynamics. It has been offered various solutions; stability, numerical dissipation, non-physical oscillations are some of the main problems with these methods. A suitable numerical method is often chosen according to the physical properties this equation represents.

Most marbling paints are immiscible, that is, they do not diffuse or mix with each other. Sharp interface lines between the paints is a typical characteristic of most marbling types. Advection solution should be able to preserve the sharp lines as the material is advected while maintaining the smoothness and continuity of the deformation. Thus numerical dissipation should be minimized in my model. Stability is also of concern, since the equations are applied on the fine grid, small time steps will increase the computation considerably.

In the following section, I will review the common finite difference advection methods, then I will discuss my method which is based on a semi-Lagrangian approach.

4.5.1 Finite Difference Methods for the Advection Equation

The main drawback of finite difference methods is that, even if the discretization provides good stability conditions, they are restricted by the step size as the equations are allowed to evolve one stencil at a time by definition. On the other hand, flux based discretization of finite differences allows good approximation and conservation properties. Lack of conservative property in advection causes material loss. In my simulation priority will be given to the visual aspects.

Conservative form of the advection equation in one dimension, which is obtained by integrating the equation over $[t^n, t^{n+1}]$ and $[x_{i-1/2}, x_{i+1/2}]$, can be written in discrete form as follows:

$$C_i^{n+1} = C_i^n - (\Delta t/\Delta x)(f_{i+1/2} - f_{i-1/2}), \tag{4.102}$$

$$C_i^n = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} C(x., t)dx, \quad f_{i+1/2} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(x_{i+1/2}, t)dt, \tag{4.103}$$

where $f_{i+1/2}$ is the numerical flux.

In finite differences the order of the approximation is measured by the number of stencils involved. Wider stencils indicate higher order accuracy [104]. CFL (Courant-Friedrichs-Levy [104]) number which is usually given by, $CFL = \frac{|U|\Delta t}{\Delta x}$, is determined according to the stencil range. (CFL condition determines the stability range based on the domain size of the finite difference and continuous forms of the PDEs.) The standard first order finite difference methods such as upwind-difference and Lax-Friedrich method [104] have the stability condition, $CFL \leq 1$, however they suffer from too much numerical dissipation. First order upwind-difference is given by:

$$f_{i+1/2} = \begin{cases} UC_i & a_{i+1/2} > 0 \\ UC_{i+1} & a_{i+1/2} < 0 \end{cases}, \tag{4.104}$$

where $a_{i+1/2}$ represents a characteristic function which can be defined as $\frac{U\Delta t}{\Delta x}$.

In higher order finite difference schemes dissipation is decreased but the high order polynomial interpolation they use lead to oscillations in large gradient regions. Second order Lax-Wendroff scheme [104] which has the stability condition, $CFL < 0.5$, is given by:

$$f_{i+1/2} = \frac{1}{2}[a_{i+1/2}(C_{i+1} + C_i) - |a_{i+1/2}|(C_{i+1} - C_i)]. \tag{4.105}$$

Application of the artificial diffusion in high order methods to overcome this problem causes the results to smear out in smooth regions. Other high resolution methods which are based on the flux limiting idea such as flux-corrected transport and Total Variation Diminishing (TVD)[104] schemes apply the artificial diffusion only in regions where it is necessary: they usually use low order approximations at discontinuities and high orders in smooth regions. Flux corrected transport, applies an antidiffusive flux on low order terms as a corrective flux in smooth regions. TVD method is based on a similar idea, only stronger conditioning is imposed to control the non-oscillatory behaviours.

Essentially Non-Oscillatory (ENO) [133] methods are also based on the flux limiting idea, they provide good non-oscillatory behaviours with high order accuracy. ENO scheme uses an adaptive technique based on selecting the smoothest stencil according to the material derivatives, in order to approximate the flux with high degree polynomials while preventing the oscillations. This adaptive interpolation can be combined with different flux types to discretize the advection equation. Since adaptivity makes it sensitive to round-off errors and costly due to the decision process, ENO technique was improved in Weighted Essentially Non-oscillatory (WENO) [133] method, by using a weighted combination of stencils. In WENO, the weights are designed according to the smoothness of the region, they are minimized at discontinuities to prevent the oscillations and optimized in smooth regions to provide high order accuracy. A third order WENO approximation uses a weighted combination as follows:

$$f_{i+1/2} = \hat{w}_0 f^0_{i+1/2} + \hat{w}_1 f^1_{i+1/2} + \hat{w}_2 f^2_{i+1/2}, \tag{4.106}$$

$$f^0_{i+1/2} = \frac{1}{3}F_{i-2} - \frac{7}{6}F_{i-1} + \frac{11}{6}F_i,$$

$$f^1_{i+1/2} = -\frac{1}{6}F_{i-1} + \frac{5}{6}F_i + \frac{1}{3}F_{i+1},$$

$$f^2_{i+1/2} = \frac{1}{3}F_i + \frac{5}{6}F_{i+1} - \frac{1}{6}F_{i+2},$$

where $f_{i+1/2}$ represents one of the components in any flux method and F represents the flux function, for example it can be defined linearly as $F = UC$. The coefficients in Eqn. 4.106 are obtained from the ENO interpolation polynomial functions [133] and the weights are defined as follows:

$$s_1 = \frac{13}{12}(F_{i-2} - 2F_{i-1} + F_i) + \frac{1}{4}(F_{i-2} - 4F_{i-1} + 3F_i), \tag{4.107}$$

$$s_2 = \frac{13}{12}(F_{i-1} - 2F_i + F_{i+1}) + \frac{1}{4}(F_{i-1} - F_{i+1}),$$

$$s_3 = \frac{13}{12}(F_i - 2F_{i+1} + F_{i+2}) + \frac{1}{4}(3F_i - 4F_{i+1} + F_{i+2}),$$

$$\hat{w}_i = \frac{w_i}{\sum_j w_j},\ w_1 = \frac{0.1}{(s_1 + \hat{\epsilon})^2},\ w_2 = \frac{0.6}{(s_2 + \hat{\epsilon})^2},\ w_3 = \frac{0.3}{(s_3 + \hat{\epsilon})^2},$$

$$\hat{\epsilon} = 10^{-6}\max(F^2_{i-2}, F^2_{i-1}, F^2_i, F^2_{i+1}, F^2_{i-2}) + 10^{-99},$$

where $\hat{\epsilon}$ is defined to prevent division by zero.

The CFL condition for ENO schemes is generally given as $\frac{(|u|+|v|)\Delta t}{\Delta x} \leq \frac{1}{2}$ [133]. ENO method can become computationally very expensive especially when the order is increased. In addition, despite its high level approximation properties at discontinuities, ENO still produces some diffusion which becomes more apparent when the fluid is highly mixed and obviously the increase in the number of steps due to small time step restriction makes this condition worse.

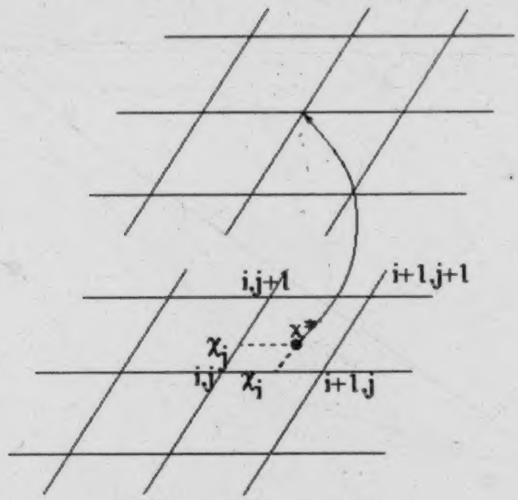


Figure 4.11: Integration of semi-Lagrangian advection.

4.5.2 Semi-Lagrangian Advection

Semi-Lagrangian methods [137] are based on the idea that the advected element is constant along the characteristic curve (Eqn.4.108) of the hyperbolic equation.

$$\frac{d\mathbf{X}(t)}{dt} = \mathbf{U}(\mathbf{X}, t) \tag{4.108}$$

In pure Lagrangian form, advection is represented by $\frac{DC}{Dt} = 0$ and mesh points are advected along the trajectory of the characteristic curve. In Semi-Lagrangian approach, information is propagated along this curve for one step on the Eulerian grid. Solution is obtained by integrating the equation over this Lagrangian trajectory in time and space. In other words, given the current values of C at time t^n, the value of a point at position $\mathbf{x}$ at the next time step, t^{n+1}, can be obtained from the backtracked position $\mathbf{x}^*$:

$$C^{n+1}(\mathbf{x}) = C^n(\mathbf{x}^*). \tag{4.109}$$

When $\mathbf{x}^*$ is in between the grid points, C values are obtained by interpolation (Figure 4.11).

When the traced position falls out of the borders, boundary conditions can be defined in a way to obtain the value from where the trajectory intersects the boundary (Figure 4.12):

$$C^{n+1}(\mathbf{x}) = C^n(\mathbf{x}^*_\Omega). \tag{4.110}$$

In the simplest first order version, which is also known as the *modified method of characteristic* (MMOC), (when forward tracking is used it is called the *method of characteristics*), the characteristic integration is done by the Euler method,

$$\mathbf{x}^* = \mathbf{x} - \mathbf{U}\Delta t, \tag{4.111}$$

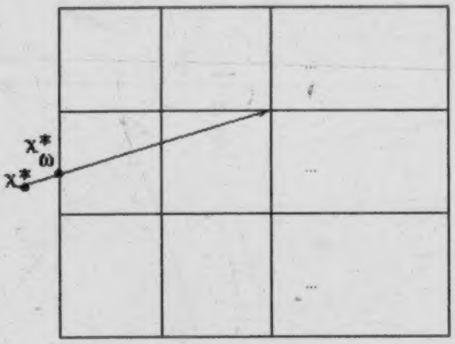


Figure 4.12: Semi-Lagrangian advection external boundary condition.

and C is obtained by bilinear interpolation:

$$C(\mathbf{x}^*) = \chi_i \chi_j C_{i+1,j+1} + (1-\chi_i)\chi_j C_{i,j+1} + (1-\chi_j)\chi_{i+1} C_{i+1,j} + (1-\chi_i)(1-\chi_j) C_{i,j}. \quad (4.112)$$

The main advantage of the Semi-Lagrangian approach is its unconditional stability. Note that, there is a consistency condition for the first order version which is $\Delta t \leq b\Delta x$. For this reason, it has been the most popular choice for the solution of advection equations in computer graphics applications [37, 120, 140]. The main drawbacks of this method are numerical dissipation, especially when low order interpolations are used, and lack of conservative property. In order to improve conservation, Roache [127] proposed to combine MMOC with finite volume flux limiters such as TVD and ENO. This method, which is called flux-based MMOC, provides higher order accuracy with local and global conservation while preserving the unconditional stability of the semi-Lagrangian approach. However this approach is computationally demanding due to complex processes required such as velocity reversals and backtracking of the volume elements in multi-dimensions. Furthermore, accuracy can be improved by using high order time integration and spatial interpolation methods. Numerical dissipation is mainly due to interpolation error. I used a first order method for the time integration as described above. On the other hand, since numerical diffusion directly affects the visual quality I used a higher order method for the spatial interpolation as discussed next.

4.5.2.1 Spatial Interpolation

The spatial approximation of the concentration values, as mentioned previously, constitutes a regular data interpolation based on a rectangular grid. Most of the criteria (such as order of accuracy, continuity etc.) given in Section 4.4 for the irregular data approximation also apply here.

The spatial interpolation for the semi-Lagrangian advection solution can also be considered as an image resampling problem. However in advection the deformation process is repeated continuously thus some artifacts which can be tolerated for one step might become an issue as the errors start building up after several steps. The image properties also play a role, for example the effect of continuity, that is the derivative information in intensity gradient, can be different in real images from the intensity gradient of lines and curves. For this reason in most image based studies [32, 112] only first order continuity is considered sufficient in reconstruction since human eye can't detect higher continuities. However in the marbling simulation, I am also interested in the deformation of lines as well as real images. Thus I will try to apply the highest continuity level possible in the interpolation method. The main problem with the high order interpolation methods is that the high degree polynomials they use might show oscillatory behaviours as they are enforced to satisfy the high order continuity condition at the nodal points. Consequently, cubic methods have been the most popular choice in both advection and image based studies.

Various studies [112, 151] have been done to compare the existing methods and to determine the optimum reconstruction technique. Here I will only consider the piecewise polynomial approximation and concentrate on different forms of splines which provide the optimum solution (according to many studies). Among the splines, I will compare Catmull-Rom splines [32, 151], Hermite cubic interpolation and B-splines which have been widely used in image interpolation.

Let's take the following convolution based general interpolation formula:

$$F(x,y) = \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_{ij} k(x-i) k(y-j), \tag{4.113}$$

where k indicates the kernel and λ_{ij} represent the interpolation coefficients.

Catmull-Rom splines have $C1$ continuity and can be represented by the following local kernel which is also called the Key's [151] function:

$$k(t) = \begin{cases} (a+2)\left|t\right|^3 - (a+3)\left|t\right|^2 + 1 & 0 \le |t| < 1 \\ a\left|t\right|^3 - 5a\left|t\right|^2 + 8a\left|t\right| - 4a & 1 \le |t| < 2 \\ 0 & 2 \le |t| \end{cases}. \tag{4.114}$$

Here $a = -0.5$ gives the optimum results (as proved in [112]). This is an interpolating kernel, that is, the coefficients represent the data points, $\lambda_i = f_i$.

Hermite cubic interpolation is also a spline approximation. It is considered close to Catmull-Rom splines with the main difference being in the definition of the slopes: Hermite interpolation uses tangent information. Catmull-Rom splines provide $C1$ continuity and it has the practical advantage of separability, that is, the interpolation can be implemented in separate passes in each dimension. This is computationally more efficient ($O(2N)$) compared to two-dimensional convolution ($O(N^2)$). Hermite cubic interpolation has also been popular in semi-Lagrangian based CFD studies [165] and lately it has been used in computer graphics [43] by Fedkiw et.al. to improve the previous method [140] which was based on bilinear interpolation. Although it can also be represented in the general interpolation formulation (as in Eqn. 4.113) a more convenient representation of the cubic Hermite interpolation in one dimension is written as follows [43]:

$$F_j(x) = \sum_{i=0}^{3} \bar{\lambda}_i (x-j)^i, \tag{4.115}$$

where the coefficients based on the derivative information are given by:

$$\begin{aligned}\bar{\lambda}_0 &= f_j,\\ \bar{\lambda}_1 &= \Delta_j,\\ \bar{\lambda}_2 &= 3\Delta_j^+ - 2\Delta_j - \Delta_{j+1},\\ \bar{\lambda}_3 &= \Delta_j + \Delta_{j+1} - 2\Delta_{j+1}^+,\end{aligned} \tag{4.116}$$

with the derivatives based on centered and forward differences defined as:

$$\Delta_j = \frac{(f_{j+1} - f_{j-1})}{2}, \quad \Delta_j^+ = f_{j+1} - f_j. \tag{4.117}$$

Finally, B-splines use polynomial basis functions to join the data points with an order determined according to the continuity desired. The basis function (i.e. the approximation kernel in the general interpolation formula given in Eqn.4.113) for cubic B-splines is defined as [153, 32]:

$$k^3(t) = \begin{cases} 0.5\,|t|^3 - |t|^2 + \frac{2}{3} & 0 \leq |t| < 1 \\ \frac{(2-|t|)^3}{6} & 1 \leq |t| < 2 \\ 0 & 2 \leq |t| \end{cases}. \tag{4.118}$$

Note that, this is not an interpolatory kernel; imposing the interpolation constraint which requires the approximation to pass through the data points results in a global matrix representation for determining the coefficients λ_{ij}. If solved in matrix form this could be a very expensive computation, however making use of the recursive property of the basis functions and the coefficients in B-splines, Unser et al. [153, 154, 155] introduced a much more efficient way of determining the coefficients with recursive filters. B-spline basis functions are defined recursively as follows:

$$k^n = k^{n-1}k^0, \tag{4.119}$$

where n represents the spline degree. According to Unser et al.'s method [153, 154, 155] the recursive algorithm for the coefficients are given by:

$$\lambda^+(i) = x(i) + z(j)\lambda^+(i-1), \quad (i = 1, .., N-1), \tag{4.120}$$

$$\lambda^-(i) = z(j)(\lambda^-(i+1) - \lambda^+(i)), \quad (i = N-2, .., 0), \tag{4.121}$$

where $x(i)$ represents the data values, N indicates the number of points in one dimension, and $z(j)$ denotes one of the n poles of the basis function. These equations are repeated for each pole j, and then for each dimension to obtain the multidimensional values by the help of separability. (Details of this technique can be found in [153, 154, 155].)

One way to evaluate an interpolation method is to consider its approximation to the ideal sinc function. (Sinc function [151, 32] is considered not realizable due to its infinite support and low decay rate.) B-splines approximate the sinc function, as their degree increases, while decreasing the interpolation error [151].

4.5.2.1.1 Experimental Results In order to test the methods, I used the single vortex velocity function which is commonly used in advection tests [6]:

$$\begin{aligned} u &= -2\sin^2(\pi x)\sin(\pi y)\cos(\pi y) \\ v &= 2\sin^2(\pi y)\sin(\pi x)\cos(\pi x) \end{aligned} \tag{4.122}$$

Figure 4.13 illustrates a comparison between bilinear and cubic Hermite methods and Catmull-Rom splines and B-splines. Among the cubic methods Hermite and Catmull-Rom splines provide very close results, on the other hand cubic B-splines are superior to both due to their $C2$ continuity. The fact that the checkerboard pattern contains both vertical and horizontal information helps preserving the level of detail when the image is highly distorted. An interpolation method is challenged when the image is highly detailed and sometimes when its features do not provide enough information to carry the level of detail to advanced stages of deformation. The effect of approximation error is obviously more effective when the resolution of the patterns is increased. In order to test the performance of the interpolation method in such a case, I used an image with relatively a high intensity of horizontal lines. Figure 4.14 shows the results from cubic, 4th and 5th order B-splines; this illustrates how more details can be preserved as the order (continuity) of the method is increased.

Unless an infinite degree interpolation is used information loss during the deformation is inevitable. Besides B-splines are not free from oscillations. Since the oscillations became more prominent after the 5th degree this was the highest level applied. The computation time for the 5th degree was much less than the cubic convolution (Key's method) because of the efficiency of the recursive algorithm.

In my opinion, the next step to improve the level of detail in advected images is to use multiresolution interpolation techniques. The information that is lost during the interpolation can be somehow extrapolated by analyzing the data at different scales. Multiresolution splines, wavelets or ENO interpolation could be some of the alternatives. In ENO studies which were proposed for the discretization of conservation laws, an interpolation method based on the same idea was also provided [133]. In this nonlinear interpolation method the adaptive selection is defined according to the singularities in the data; the selection aims at preserving the discontinuities in the smoothing process by using the derivative information. An application of this method based on a multiscale analysis have been proposed in [134]; although it provides convincing results for curves, or geometric objects, multidimensional implementation of this technique can be problematic due to the difficulty in handling the shocks in multidimensions, that is, the discontinuities at different angles. With either ENO or other interpolation methods such as wavelets or splines, multiresolution analysis would be much more costly than direct interpolation.

4.5.2.2 Sharp Interface Transformation

I propose an alternative solution to the dissipation problem in the semi-Lagrangian advection which uses an interface function in the advection equation to control the interface sharpness. Initially the concentration values, C , are transformed by this function and these transformed values are advected with the flow instead of the real values. Then, the real values are obtained from the inverse of this transformation function which actually determines the characteristics of the interface. Figure 4.15 shows the shape of the inverse interface function where the slope defines the sharpness of the interface. A similar method has been used in CFD by Yabe et.al. [165] to increase the interface

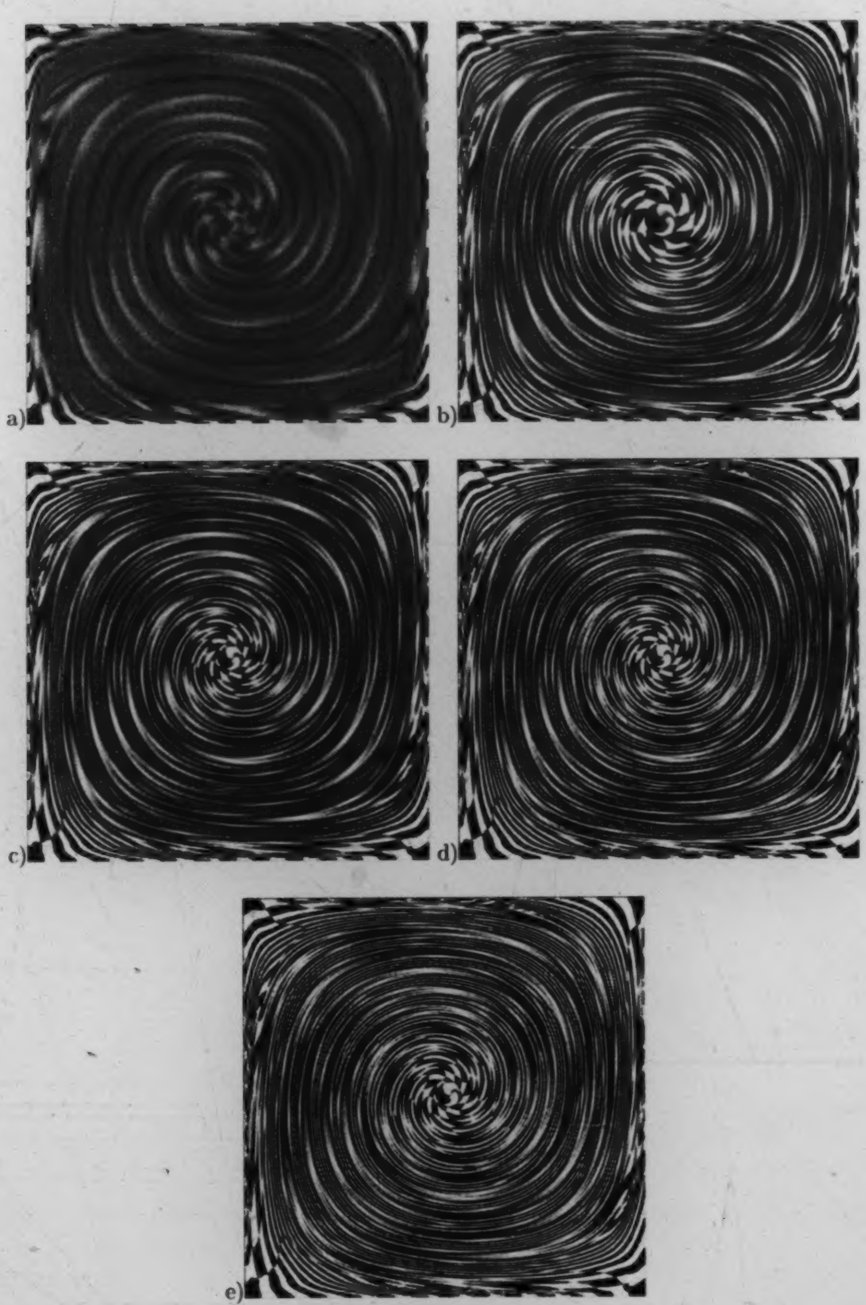


Figure 4.13: **Semi-Lagrangian advection spatial interpolation: a) bilinear, b) Hermite cubic, c) Catmull-Rom splines, d) cubic B-spline, e) 5th order B-spline.**

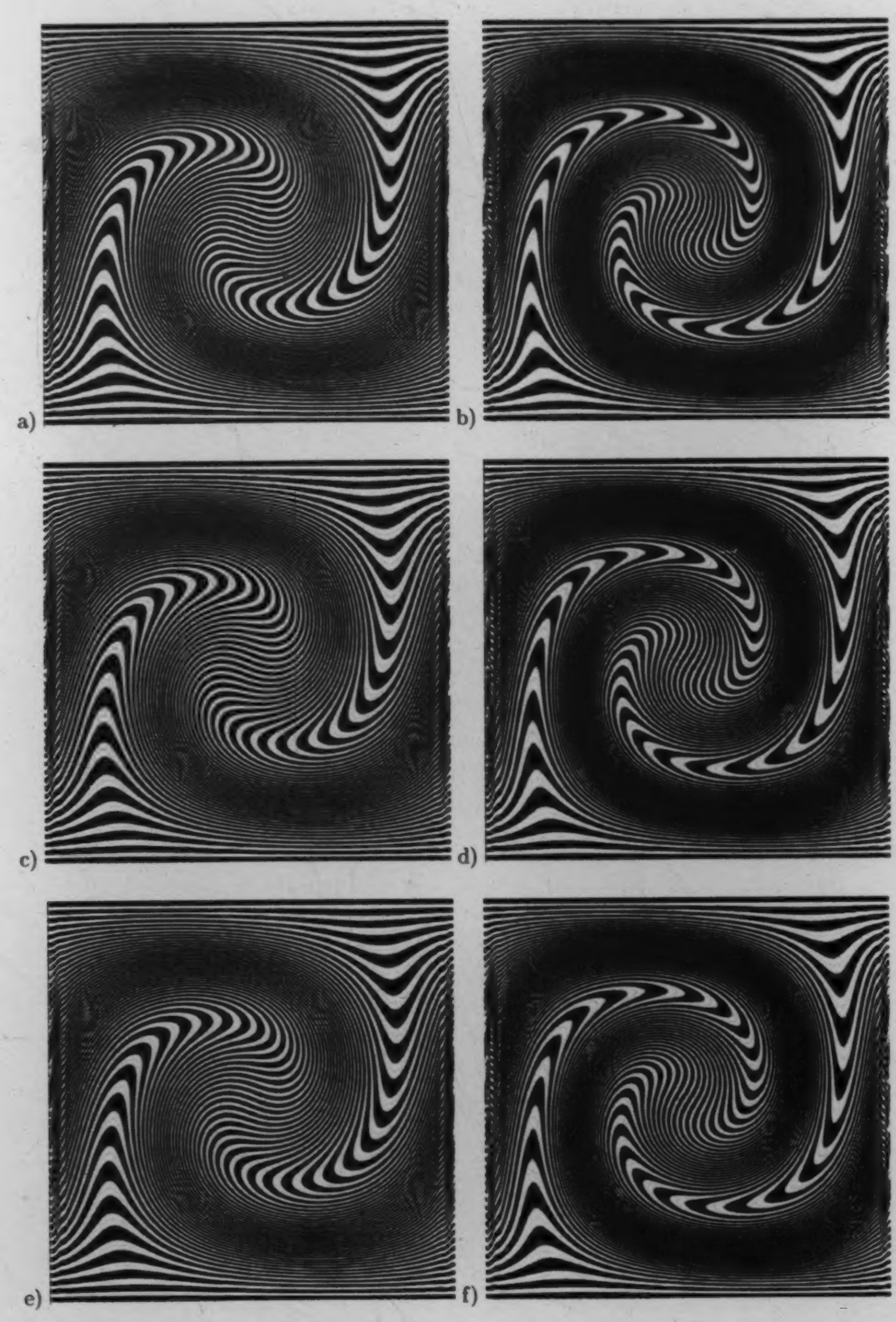


Figure 4.14: Two stages from the deformation of an image with higher level of detail using B-splines with degrees: a-b) 3, c-d) 4, e-f) 5

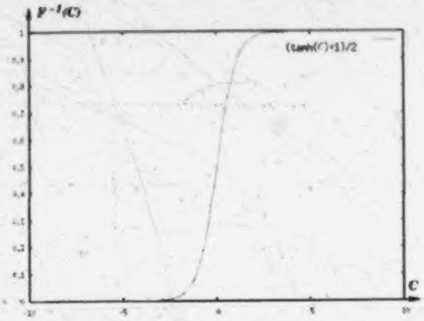


Figure 4.15: Interface function.

sharpness and to create artificial surface tension. Also note that, a level set based method has been proposed recently in computer graphics to prevent the blurring in semi-Lagrangian approach however this method is computationally expensive as it requires level set reinitialization. Current model is efficient especially for marbling simulation purposes. I defined the following transformation function based on a hyperbolic tangent:

$$F^{-1}(C) = 0.5(\tanh(C) + 1.0), \tag{4.123}$$

$$F(C) = 0.5\log(\frac{1.0 + 2.0 * C - (1.0 - \hat{\epsilon})}{1.0 - ((2.0 - \hat{\epsilon})C - 1.0)}). \tag{4.124}$$

The parameter $\hat{\epsilon}$ is included to prevent division by zero and it is defined as $\hat{\epsilon} \leq 0.0001$. Figure 4.16b shows how the interface sharpness is maintained with the application of the transformation function. It should be pointed out that even though this method resolves the dissipation problem in terms of interface thickness, the information loss due to dissipation, and consequently the level of details in the deformation still depends on the accuracy of the interpolation. I should also emphasize that sharp interface transformation removes the oscillations in high order interpolations.

From a different perspective, application of this transformation function can be considered as adding a reaction term in the advection equation. Solution of advection with nonlinear reaction terms by using standard techniques can be problematic. In [97], Kojouharov et.al. presented a nonstandard solution to the advection equation with nonlinear reaction terms, which is actually very similar to this transformation idea. In this method, the reaction term, $R(C)$, is represented by, but not limited to, polynomials, $q(C)$:

$$\frac{\partial C}{\partial t} + \mathbf{U} \cdot \nabla C = R(C). \tag{4.125}$$

When the reaction term is formulated as $R(C) = \frac{q(C)}{q'(C)}$, Lagrangian form of the equation can be written as:

$$\frac{Dq(C)}{Dt} = q(C). \tag{4.126}$$

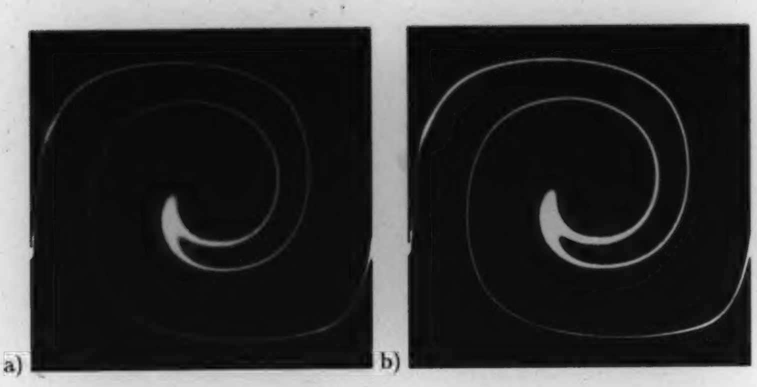


Figure 4.16: Advection with a) bilinear interpolation b) sharp interface transformation with bilinear interpolation.

This implies that the solution can be obtained by advecting the function $q(C)$ with the semi-Lagrangian approach as described above:

$$q^{n+1}(x, C) = q(x^*, C), \tag{4.127}$$

and the real C value can be obtained from the inverse function,

$$C^{n+1} = q^{-1}(C^n). \tag{4.128}$$

4.5.3 Reaction

Most paints used in marbling are chemically reacting. For example well-known vein patterns (as shown in Chapter 2) are formed as a result of the reaction between different types of pigments. Lately contemporary marbling techniques have been exploiting the chemical properties of the paints [79, 109]. For example the spiral patterns used in [79] which are formed as a result of chemical interface bifurcation can be modeled by several different reaction-diffusion models such as Ginzburg-Landau model. Reaction-diffusion patterns (including the spiral pattern) were also investigated in computer graphics [161], these studies were based on Turing's original reaction-diffusion model. In the future it would be interesting to study various chemical properties of marbling paints. Here in order to simulate the continuous extension of a material I use a bistable reaction system based on a Fitzhugh-Nagumo model which is classified under excitable media. Reaction term included in the advection equation is given as follows:

$$\frac{dC}{dt} = DaR(C), \tag{4.129}$$

$$R(C) = C(a - C)(C - 1), \tag{4.130}$$

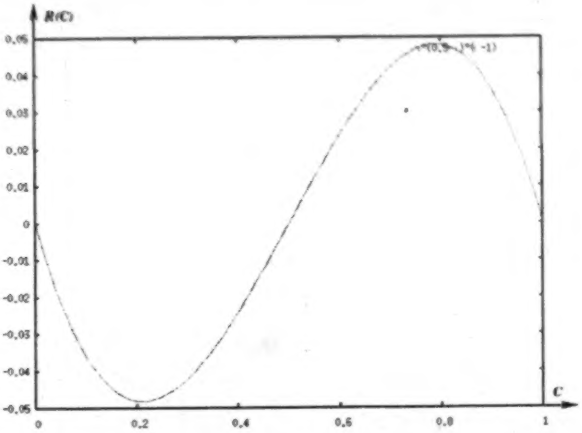


Figure 4.17: Bistable reaction term.

with $0 < a < 1$. Here R indicates the reaction term and Da is known as the Damkohler number and it represents the strength of the reaction. I discarded diffusion, although it provided smoothness it also caused loss of detail. A bistable system is a system with two stable solutions. Evolution of the equation involves the interaction of two states, and it ensures that the system converges to one of these two states. (Behaviour of this model under chaotic mixing has been studied in physics studies [119].) In this model, depending on the reaction rate, either the advection or the reaction dominates, and either way the system approaches to one of the two states. When Da is below a critical value, the system runs towards $C = 1$ state and the material extends, otherwise it returns to $C = 0$. Also, only the concentration values above a survive and eventually reach 1 and the values below a tend to disappear. (Figure 4.17 shows the characteristic reaction function for $a = 0.5$.) These parameters control the material extension as it is advected. Figure 4.18 shows the effect of the reaction term on the test case.

4.6 Interface Dynamics

The interface between two liquids is in a different physical state than the bulk of the fluid. Interface is subject to interfacial forces such as surface tension and under convection it goes through topological transitions such as break-up and coalescence. These processes are affected by viscosity, pressure and velocity as well as the chemical and rheological properties of the fluids such as elasticity. The complexity of these events which occur at different scales makes it difficult to study the dynamics of the interface.

In order to examine the influence of interfacial forces let us consider a droplet under convection. As the fluid is mixed droplets are stretched and deformed and eventually they get thinner and break into smaller droplets. Smaller droplets on the other hand may coalesce into larger drops

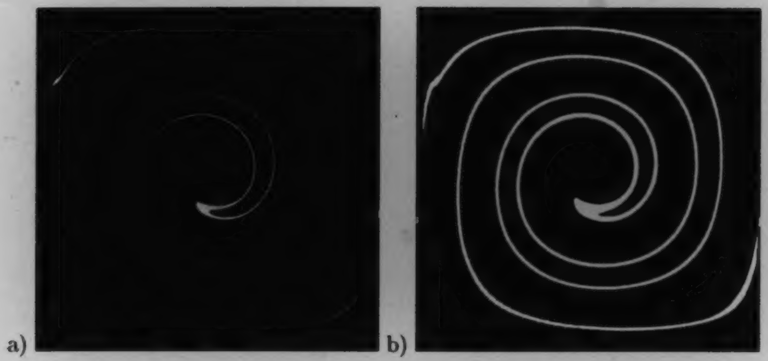


Figure 4.18: Advection: a) without reaction, and b) with the reaction term with $a = 0.3$.

when they come into contact with each other. Capillary number, Ca is a characteristic value of the flow representing its deformation properties. It is generally defined as the ratio of viscous forces to interfacial forces as follows:

$$Ca \approx \frac{\dot{v}\nu R}{\sigma_s}, \tag{4.131}$$

where $\dot{v}$ and ν represent the flow shear rate (velocity gradient) and viscosity respectively, R indicates the initial radius of the droplet (or the striation thickness) and σ_s denotes the surface tension. This representation shows that the resistance of a droplet to the flow increases as its size decreases. If $Ca \gg 1$ the viscous forces dominate and surface tension might be negligible, otherwise deformed droplets break into smaller ones in shorter time periods and increasing sizes as Ca decreases. Coalescence takes place when the droplets come into close contact; if they are close enough a liquid film is forced between them and if this film becomes thick enough it breaks at thin points and the drops coalesce due to van der Waals forces. Interface thickness or surface tension of the phase field, interface mobility and droplet size as well as the viscous forces are the main factors that affect this event. Coalescence is more likely to happen at low shear rate and at high interface mobility and surface tension.

4.6.1 Numerical Modeling

Interface modeling will be analyzed in 2 groups.

1. Front tracking or free boundary methods.
2. Front capturing methods.

4.6.1.1 Front Tracking (Free Boundary) Methods.

Conventionally surface tension is described as a geometric surface property. It is represented as a jump in the stress tensor normal to the interface by appropriate boundary conditions which are generally formulated as follows:

$$(\mathbf{T}_1 - \mathbf{T}_2)\cdot\mathbf{n} = \frac{1}{Ca}\mathbf{n}(\nabla_{\mathbf{n}}\cdot\mathbf{n}). \tag{4.132}$$

Here $\nabla_{\mathbf{n}}$ represents the interface curvature and the left side of the equation represents the discrete changes in the stress tensor, $\mathbf{T}$, on the interface.

This classical approach resolves the interface issue as a free boundary problem, in other words, the interface is tracked as a free boundary. This is accomplished by remeshing, that is, moving the grid along with the interface in accordance with the given boundary conditions. Basically there are two drawbacks of this system. The first one is the numerical difficulties caused by the discontinuous representation of the boundary. The second drawback is the complexities caused by the adaptive grid technique; continuously remeshing the grid can become computationally expensive and complicated especially in large deformations, hence this method might have difficulties in handling topological transitions.

There is also the pure Lagrangian way of tracking the interface using particles. This technique is usually more appropriate for modeling free surface flows rather than surface tension effects. This approach preserves the interface sharpness however with the particles one always has to deal with the N-body problem to locate the neighbouring particles as they move independently. Besides keeping the continuity of the surface as the particles part or overlap can be problematic.

4.6.1.2 Front Capturing Methods

In order to remedy some of these difficulties that occur in front tracking methods, front capturing techniques have been proposed. In these methods, interface is represented by a phase function which distinguishes different phases. This phase function (C_ϕ) is advected by the flow on a fixed grid:

$$\frac{\partial C_\phi}{\partial t} = -\mathbf{U}\cdot\nabla C_\phi. \tag{4.133}$$

Evidently from this Eulerian description, enforcement of the boundary conditions as a discontinuity across the interface is no longer required. In Volume of Fluid (VOF) [72] method this phase parameter describes the fractional volume of fluid in each cell and it is represented by a line in that cell. The position and direction of the line is adjusted according to this fractional value with an appropriate algorithm to reconstruct the interface. Usually it is difficult to maintain the continuity and smoothness of the interface with this method due to numerical diffusion and the line representation. Besides the reconstruction algorithm can be computationally expensive.

In computer graphics, lately Hong et.al. used a VOF method in [73] to simulate the motion of air bubbles in a liquid. They used the marching cubes algorithm for surface construction and they also combined the VOF method with particles in some applications to add more detail. Even though the marching cubes algorithm improved the smoothness to an extent, the discontinuities which is a major drawback of the VOF method due to its line representation, are still evident in the results.

In the Continuum Surface Force (CSF) [13] method, interface is represented by a color function indicating the variation of density at the interface as a transition region. This smooth representation in CSF eliminates the discontinuity problem that occurs in VOF method. However numerical diffusion can cause the interface to smear. In order to maintain the interface continuity and sharpness, the color function needs to be redistributed. The surface tension force in all the front capturing methods in general, depends on the interface curvature and normal, and the jump in the phase parameter. Surface tension in CSF which is represented as a volumetric force that changes smoothly across the interface is given by:

$$F_{SV} = \sigma_s \kappa \mathbf{n} \delta(C_\phi), \tag{4.134}$$

where σ_s denotes the surface tension coefficient and κ represents the interface curvature based on the surface normal $\mathbf{n}$ as follows:

$$\kappa = -(\nabla \cdot \mathbf{n}) = \nabla \cdot (\frac{\nabla C_\phi}{|\nabla C_\phi|}). \tag{4.135}$$

The interface delta function δ is given by:

$$\delta(C_\phi) = \frac{|\nabla C_\phi|}{[C_\phi]}, \tag{4.136}$$

where $[C_\phi]$ represents the jump, that is, the difference in C_ϕ values across the interface.

Level set method [147] is based on a similar idea; in this method the phase parameter is represented by a level set function which is defined as a signed distance to the interface as follows:

$$C_\phi(r) = \begin{cases} > 0 & r \in Phase\ 1 \\ = 0 & r \in Interface \\ < 0 & r \in Phase\ 2 \end{cases}, \tag{4.137}$$

where r indicates the distance. The surface tension force in the level set method is defined as in CSF, only the interface delta function is represented by a Heaviside function.

Level set function is designed to preserve the sharpness of the interface, hence smearing is not a problem in this method however as the level set is advected, signed distance property may not be maintained due to numerical diffusion. In order to restore the signed distance property, level set value needs to be reinitialized. This is an iterative and costly process [148] and also imposes a time step constraint. Alternative solutions have been proposed [53] to remove numerical diffusion effects in the level set methods by using particles. This hybrid particle level set approach makes use of the flexibility of the Lagrangian representation of particles to make up for the information loss and also to allow free, dynamic movement of the surfaces. Later this method was improved in [37] to obtain more realistic flows by extrapolating the surface velocity; the particles were actually used to correct the diffusion error in this method. These methods have been successfully used in graphics to model free surface flows. However while being very suitable for the simulation of wavy, splashy free flow of liquid surfaces, the particle level set approach is not appropriate for a more detailed image-based deformation problem. Besides as discussed previously, particle representation is not suitable for modeling continuous surface tension effects.

One of the advantages of the front capturing methods is that the topological transitions are handled naturally. Another front capturing method which has been recently coming to attention is the diffuse-interface model; this method has not been explored in graphics for fluid interface modeling to my knowledge. My approach for the interface problem, which is based on this method, will be discussed next.

4.6.2 Diffuse-Interface Model

Diffuse Interface (DI) method is relatively a new method in CFD although the underlying Cahn-Hilliard theory goes back to 1958 [15]. It is similar to the CSF and level set methods in the sense that the interface is represented by a phase parameter which is advected with the flow. However in diffuse-interface approach the phase parameter is determined by the interface physics based on mesoscale interactions. The evolution of this parameter is described by the Cahn-Hilliard equation. In diffuse-interface modeling, topological transitions are handled automatically and the application of complex algorithm such as for interface reconstruction or phase parameter reinitialization are not necessary. In addition, this method is flexible in controlling the interface features such as sharpness.

4.6.2.1 Cahn-Hilliard Theory

Cahn-Hilliard theory describes the interface dynamics by the evolution of an interface potential based on the minimization of surface free energy. This is consistent with the thermodynamic description of the interface of an isothermal fluid which depends on its free energy density [156]. The interface potential is obtained from the calculus of variations according to van der Waals theory [156], in order to minimize the free energy. This potential is given by,

$$\mu = \frac{\delta \mathcal{F}}{\delta C} = f'_\varphi(C) - \hat{\xi}^2 \nabla^2 C, \tag{4.138}$$

where C and μ indicate the phase parameter and the potential respectively. The free energy, $F = \int f dV$, is defined by the free energy density, $f = f_\varphi(C) + \frac{1}{2}\frac{1}{\xi^2}|\nabla C|^2$. In this functional, $f_\varphi(C)$ represents the bulk free energy density which is usually taken as the Ginzburg-Landau free energy function, $f_\varphi(C) = \frac{1}{4}(C^2-1)^2$, in the Cahn-Hilliard models. The gradient term is actually introduced to improve the minimization process in a way by balancing the abrupt energy changes in the phase parameter. (See also [45] for the mathematical aspects of this model and its derivation.) The parameter $\hat{\xi}$ represents the interface thickness, in other words it defines the interaction range of the interfacial activities. Hence, it is an indicator of fluid's immiscibility. The evolution of C in time is described by Cahn and Hilliard as follows:

$$\frac{\partial C}{\partial t} = M\nabla^2\mu = M\nabla^2(f'_\varphi(C) - \hat{\xi}^2\nabla^2 C) = M\nabla^2(C^3 - C - \hat{\xi}^2\nabla^2 C). \tag{4.139}$$

Here M represents the interface mobility. This representation for the evolution of the potential is designed to conserve mass. Basically this equation describes the evolution of the phase parameter as an interplay between the free energy and the gradient terms: these two terms balance each other to minimize the surface energy. The roots of the double well free energy identifies the stable and unstable regions of the solution. The roots, which are ±1 for the given energy function, identify the equilibrium solution. When the average concentration is in the region where $f''_\varphi(C) < 0$ (Figure 4.19), the system is in an unstable state which is usually referred to as the ***phase separation*** or ***spinodal*** stage. In this state, the system is susceptible to any perturbation and results in the separation of phases into the equilibrium values; during the fast evolution of this stage the gradient energy increases. In fact, Cahn-Hilliard(CH) theory was originally proposed [15] for modeling the spinodal decomposition in binary alloys. Once the phases are separated into equilibrium values then

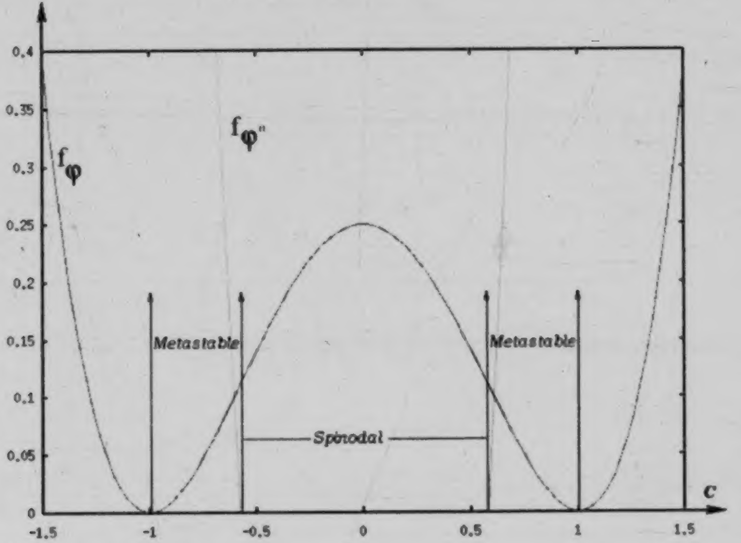


Figure 4.19: Free energy function: stability regions.

each component of the potential behaves in a way to minimize the surface energy. When the average concentration is in the intervals where $f''_\varphi(C) > 0$, the system is in a metastable state. At this stage, which is also called the *phase coarsening* stage, the interface, while working towards global minimization, is considered locally stable. Coarsening proceeds much slower than the phase separation. In fluid interface modeling I am mostly concerned with the coarsening process.

Figure 4.20 shows the evolution of these phases. The concentration values C are initialized to values in the spinodal interval (between ±0.00005) with random perturbation. Interface thickness and mobility are taken as $\hat{\xi} = 0.004$, $M \triangle t = 0.005$. Evolution of the phase parameter from Figure 4.20a to Figure 4.20b illustrate the phase separation, Figures 4.20c,d,e,f illustrate coarsening.

Topological transitions such as coalescence and breakup take place naturally according to the phase field evolution. The scale of these processes are determined by the interface thickness, that is, the interaction range. Figure 4.21 shows the coalescence of two droplets. In these examples parameters of the Cahn-Hilliard equation are given as, $\hat{\xi} = 0.0045$, $M \triangle t = 0.005$. In order for the interaction to take place, that is to allow the overlapping of the interface layers, the distance between the droplets should be proportional to the interface thickness. Finally, Figure 4.22 shows how a line, which is thin enough in terms of the interaction range, breaks into smaller droplets. Here I used a thicker interface ($\hat{\xi} = 0.025$) compared to the coalescence test just to visualize the evolution of the process better and I increased the display value of the concentration to demonstrate the layers. Otherwise, when distributed in $[0, 1]$ scale, the interface does not appear as sharp.

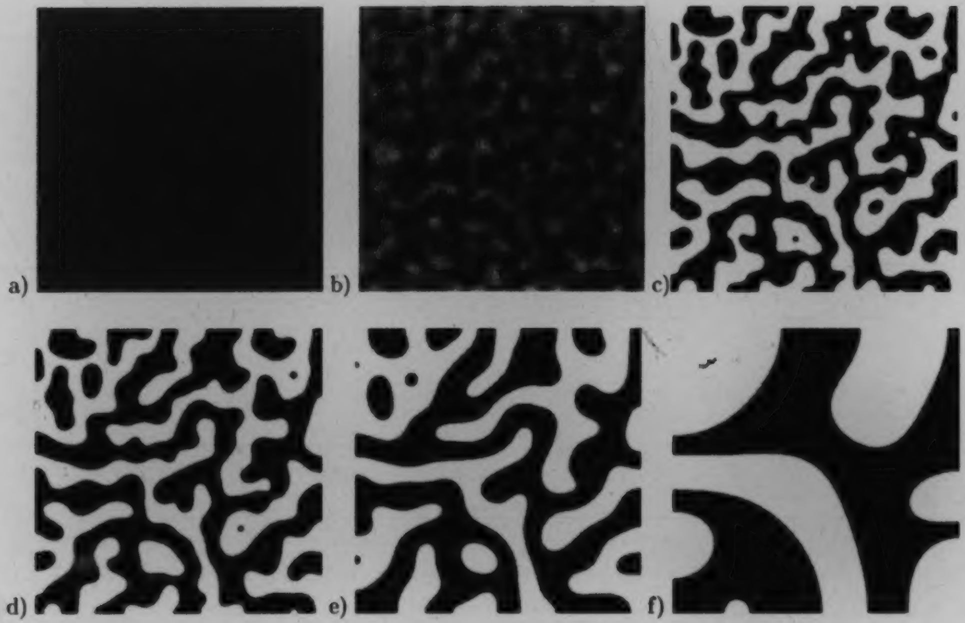


Figure 4.20: Phase separation and coarsening. a) t=0, b)t =30Δt, c) t=60Δt, d) t=210Δt, e) t=420Δt, f) t=2100Δt.

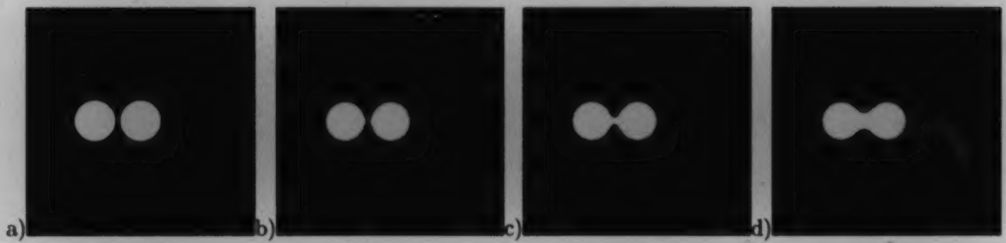


Figure 4.21: Coalescence of two droplets. a)t=0 , b) t =40Δt, c) t =55Δt, d) t =100Δt.

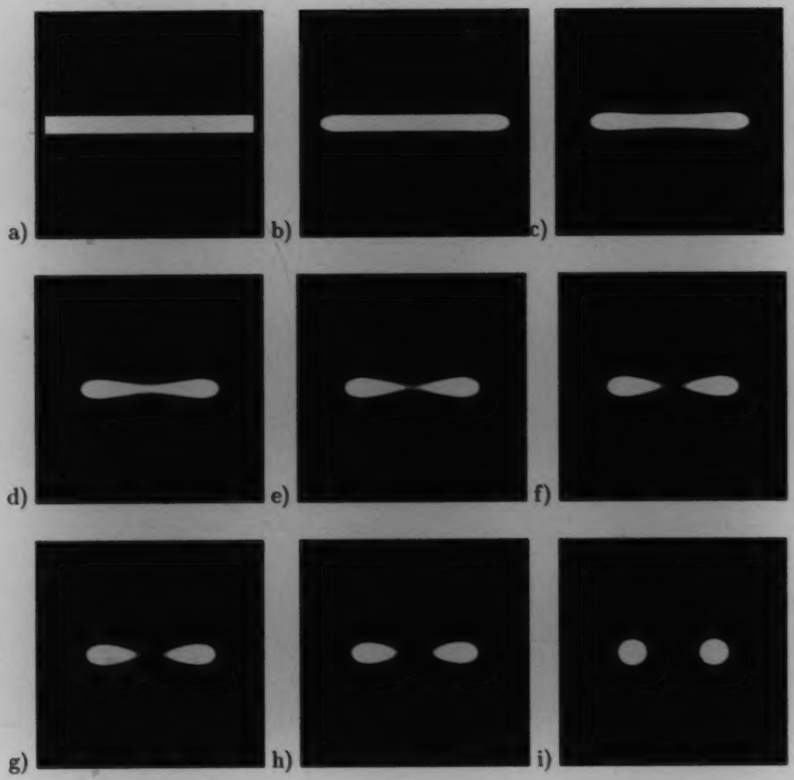


Figure 4.22: Breakup of a line into two droplets: a) t=0 , b) t $=7\Delta t$, c) t $=15\Delta t$, d) t $=22\Delta t$, e) t $=24\Delta t$, f) t $=25\Delta t$, g) t $=26\Delta t$, h) t $=28\Delta t$, i) t $=43\Delta t$.

4.6.2.2 Numerical Solution of the Cahn-Hilliard-Advection Equation

The transport model includes the advection and the Cahn-Hilliard diffusion terms:

$$\frac{\partial C}{\partial t} = -\mathbf{U}\cdot\boldsymbol{\nabla}C + M\nabla^2\mu. \tag{4.140}$$

I solve this equation with operator splitting. Operator splitting idea is based on splitting the time step of the equation so that different terms can be solved separately in several substeps. Although splitting increases the truncation error, it is adventageous as it allows the application of an appropriate method for each term. Based on this idea, Cahn-Hilliard advection equation (Eqn.4.140) can be divided into two substeps as follows:

$$\frac{C^{*} - C^{n}}{\Delta t} = -\mathbf{U}\cdot\boldsymbol{\nabla}C, \tag{4.141}$$

$$\frac{C^{n+1} - C^{*}}{\Delta t} = M\nabla^2\mu. \tag{4.142}$$

The advection part (Eqn.4.141) is solved by the semi-Lagrangian method (as described above). Numerical solution of the diffusion part, Cahn-Hilliard equation (Eqn.4.142) is difficult because of its stiff characteristics due to the 4th order Laplacian and the nonlinear terms (Eqn.4.139).

In finite difference discretization if explicit integration is used, time step condition becomes too restrictive. On the other hand, the implicit matrix solution becomes ill-conditioned because of the complexity of the finite difference formulation. As a result, spectral methods have been popular for the spatial discretization of this equation due to the convenience in using fast Fourier transform in discretization of the Laplacian terms and implicit matrix solution. I used the unconditionally stable semi-implicit discretization proposed by [39, 40]. This method is based on splitting the linear terms to achieve unconditional gradient stability. (Gradient stability means that the energy of the system decreases at each time step). I should note that this is not the only way of splitting to obtain gradient stability (a comparison for other possibilities is provided in [39]). However this scheme is designed to achieve unconditional gradient stability with less effort and easier computation while minimizing the truncation error. This linearly stabilized splitting scheme is given by,

$$C^{n+1} = C^{*n} + \Delta t(-\frac{1}{\xi^2}\nabla^4 C^{n+1} + \chi_s\nabla^2 C^{n+1}) + \Delta t(\nabla^2(C^{*n})^3 - (1+\chi_s)\nabla^2 C^{*n}), \tag{4.143}$$

with the boundary conditions:

$$\boldsymbol{\nabla}C\cdot\mathbf{n} = 0, \tag{4.144}$$

$$\boldsymbol{\nabla}\mu\cdot\mathbf{n} = 0. \tag{4.145}$$

The matrix solution, $\mathbf{Ax} = \mathbf{b}$, is obtained according to the following discretization:

$$(I + \Delta t \frac{1}{\hat{\xi}^2} L^2 - \chi_s L) C^{n+1} = DCT(C^{*n}) + \Delta t L * DCT((C^{*n})^3 - (1 + \chi_s) C^{*n}). \tag{4.146}$$

Here L denotes the eigenvalues of the Laplacian operator which can be represented by cosine basis functions based on the boundary conditions given in Eqn. 4.144, 4.145. According to this algorithm given in [40], first discrete cosine transformation of the concentration values are taken, next based on a spectral decomposition eigen values of the matrix **A** are calculated which are then divided by **b**. Finally, the real values are obtained from the inverse discrete cosine transformation of the result.

4.6.2.3 Cahn-Hilliard/Navier–Stokes Coupling

Capillary, or surface tension force is included in the momentum equation to balance the energy increase which is caused by the interface free energy. In other words, the force applied on the fluid by the interface is obtained from the energy conservation constraint. This force, F_S, depends on the phase parameter and its gradient as follows:

$$F_S = \mu \nabla C. \tag{4.147}$$

The derivation of this term based on the energy conservation idea can be found in [89]. For incompressible fluids this force can be written as [90]:

$$F_S = C \nabla \mu. \tag{4.148}$$

Another representation for the capillary stress is given by [68]:

$$F_S = \nabla \cdot (\nabla C \otimes \nabla C) = \Delta C \nabla C - \frac{1}{2} |\nabla C|^2 . \tag{4.149}$$

In the incompressible limit the gradient term can be absorbed into the pressure [68] and the capillary force in the momentum equation can be represented by:

$$F_S = -\sigma_s \Delta C \nabla C, \tag{4.150}$$

where σ_s represents the surface tension effect or the capillary number which is proportional to the interface thickness $\hat{\xi}$. The proof of the convergence of this approach to the classical sharp interface description is discussed in [87]. I chose this representation for the capillary force mainly for numerical simplicity. Since this term is discretized on the coarse grid, I used the concentration values from the fine grid locations which correspond to the fluid grid points. Using an averaged value could be an alternative to reduce the approximation error however in my experiments this did not make a big difference. In addition, I used a relatively high resolution fluid grid to obtain a better interface approximation. Despite these efforts, the oscillations in the interface force which are basically due to the errors in approximating the interface by the coarse grid cannot be prevented. In order to remove this noise I applied high level smoothing to the velocity vectors.

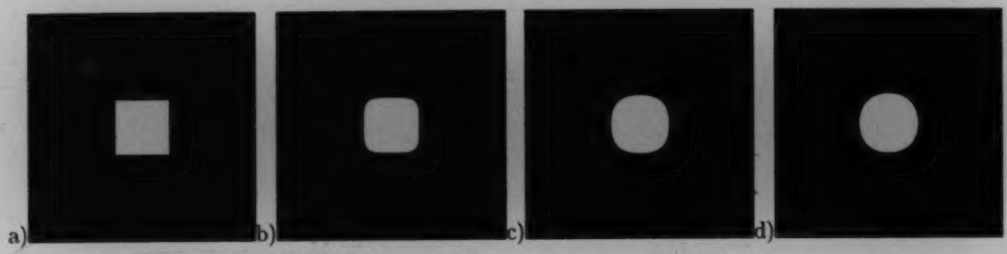


Figure 4.23: Evolution of a square with CH model: a) t=0 , b) t $=400\Delta t$, c) t $=1200\Delta t$, d) t $=2400\Delta t$

The capillary force included in the momentum equation provides the same surface effect, as obtained from the Cahn-Hilliard equation in mesoscopic level, at the macroscopic scale. It provides a faster evolution of the interfacial events at larger scales, and the interfacial effects of the Cahn-Hilliard equation are mostly dominated by this force when it is included in the momentum equation. Figure 4.23 shows the coarsening of a square in a) with only Cahn-Hilliard diffusion and in b) with the Navier-Stokes-Cahn-Hilliard coupling with the Navier–Stokes parameters, $\Delta t = 0.7, \nu = 0.2\sigma_s = 0.5$. This is a classic test case used in surface tension modeling; as a result of surface smoothing a square first turns into a rhomboid like shape and then into a circle. This type of evolution is specific to the solution of fluid equations. With the Cahn-Hilliard equation the square smoothly turns into a circle. In Figure 4.23a one can see how long this process takes with only Cahn-Hilliard equation. As the interface thickness is decreased, the evolution of this process becomes even slower. On the other hand with the Navier–Stokes solution the same smoothness and continuity can not be achieved (due to the coarseness of the approximation). In my opinion, increasing the evolution speed of the Cahn-Hilliard equation could be an interesting alternative for the solution of this problem. In this way, continuity can be guaranteed at the image resolution. However modifying the Cahn-Hilliard model for this purpose is neither trivial nor numerically easy. This issue will be discussed next.

4.6.2.4 Nonlocal Effects

Evolution of the Cahn-Hilliard equation is slow due to the local gradient based description of the interactions as it describes the interface interactions at mesoscopic levels. Although the numerical solution is unconditionally stable, increasing the time step and/or the mobility after a certain limit doesn't make any difference if not cause inaccuracy. One approach to resolve the interface speed problem is to include global attractions. This is generally done, in theory, by adding a nonlocal term in the free energy density [19, 123, 45, 56]. For example, in order to model the phase separation in incompressible systems, an additional pressure term, which is designed to enforce incompressibility, is included in the free energy [36]. This additional term introduces convective energy in the system which gives rise to nonlocal effects increasing the evolution speed. The concentration potential with nonlocal contribution can generally be given as,

$$\hat{\mu} = \mu + \int K(\|x - y\|)C(y)dy, \tag{4.151}$$

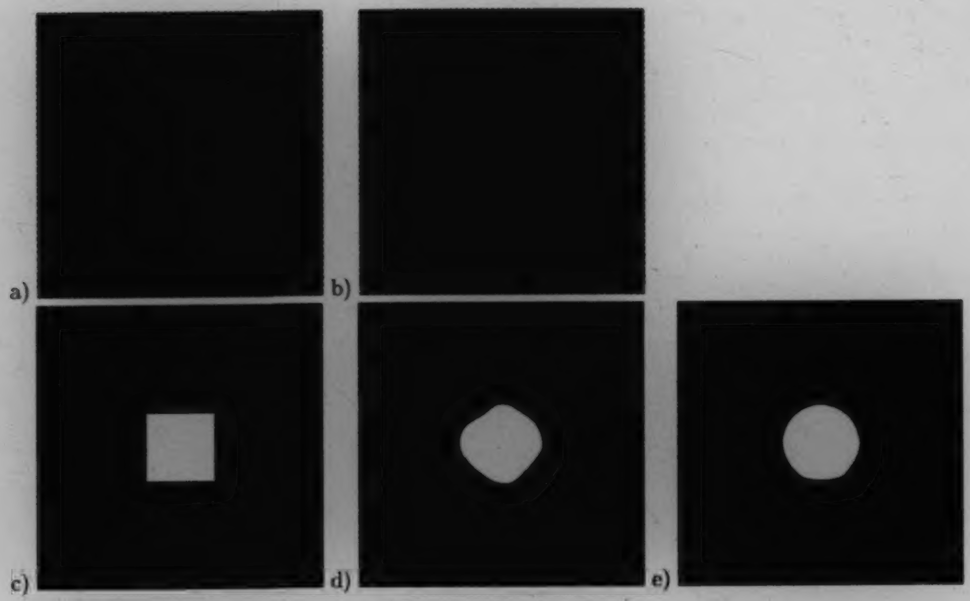


Figure 4.24: Evolution of a square in CH-NS model: a) t =5Δt, b) t =10Δt,c) t=0 , d) t =5Δt, e) t =10Δt.

where $\widehat{\mu}$ represents the modified potential and K is a distance based kernel. Note that, global elasticity is formulated in a similar way by including the elastic energy in the free energy density [56]. In that case the kernel, K, represents the elasticity coefficients which are obtained from the elasticity equation [103]. In linear, homogeneous elasticity where the coefficients are not concentration dependent, only a constant contribution can be assumed without the need for solving the elasticity equilibrium equation.

Given these, I developed a rather simplistic method (based on a linear contribution) to speed up the process in a way by also including elastic effects. I added a similar term in the free energy based on the local concentration average as given by:

$$\hat{C} = \sum_i \sum_j K_{ij} C(x_i, x_j), \tag{4.152}$$

where K represents the discrete form of a Gaussian smoothing filter. (A 7 × 7 filter area is used in the experiments.) This term is included in the split form of the equations as follows:

$$C^{n+1} = C^{*n} + \Delta t(-\frac{1}{\hat{\xi}^2}\nabla^4 C^{n+1} + \chi_s \nabla^2 C^{n+1}) + \Delta t(\nabla^2 (C^{*n})^3 - (1+\chi_s)\nabla^2 (C^{*n} + \hat{\alpha}\widehat{C})), \tag{4.153}$$

where $\hat{\alpha}$ represents an amplification factor. As evident from this implementation, linear part of the energy is modified only in the current time step component. This was to be able to use the current discretization method otherwise eigenvalue computation would be difficult. This additional term

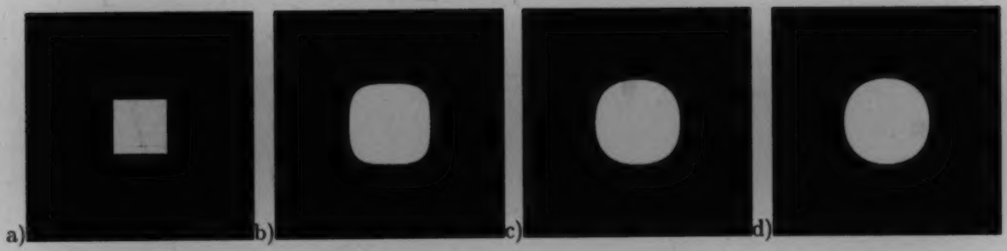


Figure 4.25: Evolution of a square in CH model with elastic energy: a) t=0 , b) t =200Δt, c) t =600Δt, d) t =900Δt.

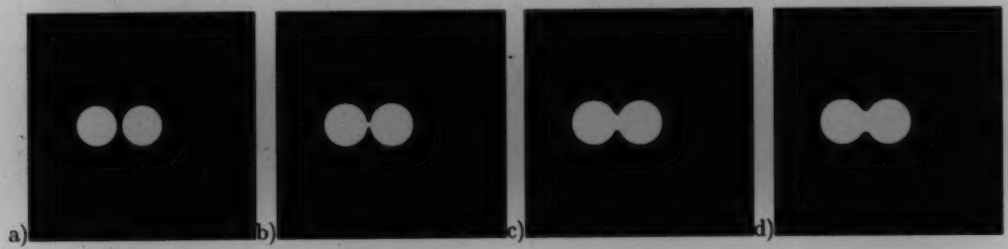


Figure 4.26: Coalescence in CH model with elastic energy: a) t=0 , b) t =70Δt, c) t =120Δt, d) t =170Δt.

increases the speed of coarsening, it also includes elastic energy in the system causing the material to be extended. Obviously mass conservation in material advection is no longer satisfied, however as mentioned before this was not a primary concern in the marbling model as also most marbling paints which have elastic properties do not comply with this rule.

I tested this method with the square example and also for the topological events. Figure 4.25 shows the relatively fast transformation of the square with the nonlocal effects, also material extension in the expansion of the circle can be observed. In these examples I used the same parameters as in the corresponding tests given above. Figures 4.26 and 4.27 demonstrate the difference in the evolution of topological events when extra energy is included: coalescence happens faster and breakup doesn't take place due to elasticity.

I would also like to mention some aspects of the simulation of polymeric solutions. Cahn-Hilliard equation with Ginzburg-Landau free energy function describes the properties of polymeric solutions to an extent, however for more realistic simulations Flory-Huggins free energy is suggested [65], which is also a double well energy similar to the Ginzburg-Landau energy. However numerical solution of the system needs to be reconsidered for different energy functionals as the linear splitting scheme might not be suitable. In fact, although these systems might provide a more realistic modeling of the physical details I doubt their possible advantage in the visual aspects of the marbling simulation. However in the future, a more advanced nonlocalization or elasticity modeling can be considered.

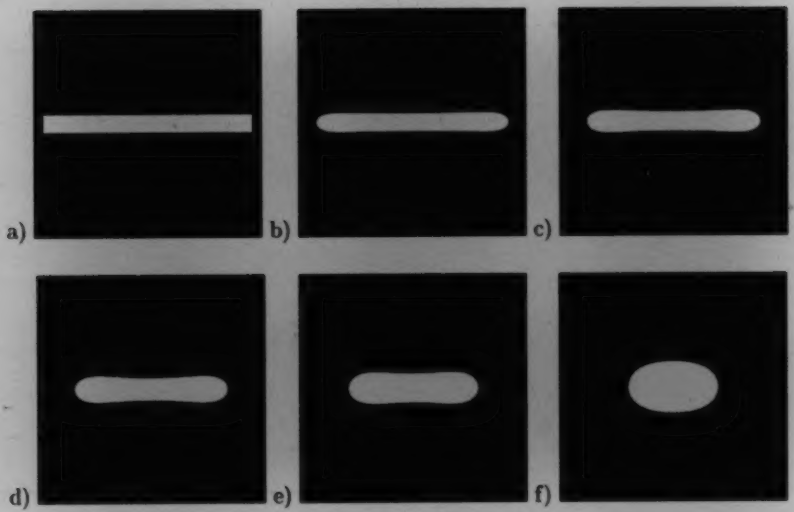


Figure 4.27: Evolution of the line in breakup example under elastic energy: a) t=0 , b) t =7Δt, c) t =15Δt, d) t =26Δt, d) t =43Δt, d) t =90Δt.

Chapter 5

Digital Marbling

This chapter presents the digital marbling model which is categorized based on an analysis of the traditional marbling techniques in terms of fluid dynamics. Marbling processes are classified according to the Reynolds number. It should be pointed out that this classification does not necessarily indicate totally independent physical processes. Reynolds number is a critical parameter in characterizing a fluid flow. It is the main criterion for determining if the flow is laminar or turbulent and it represents the ratio of inertial to viscous forces. It is usually given as:

$$Re = \frac{UL}{\nu}, \tag{5.1}$$

where L and U indicate the characteristic length and mean flow velocity respectively, and ν denotes the kinematic viscosity. In laminar flows, that is, at low Reynolds numbers, the flow is relatively smooth. As the Reynolds number increases, after a certain value, the flow becomes turbulent; viscous forces aren't strong enough to smooth out small fluctuations and the flow displays an irregular motion with eddies and vortices. In the laminar case, in very viscous marbling flows, that is when the background liquid is very thick, the paint solution behaves like an elastic material rather than a fluid and the effect of boundary is very local. This very typical form of marbling which can be observed in the comb patterns in (see Chapter 2) can be more appropriately modeled by the use of simpler mathematical systems. These comb patterns are simulated in Fractal Design Painter [54] with automatic functions. And lately [85] have defined diffeomorphism based functions to model the deformation caused by the pins of the comb. The main interest of this thesis is in marbling flows which make use of the dynamics of the fluid motion, that is unpredictable, chaotic flows. However I will demonstrate how a fluid method can also be used to model comb patterns, in a rather unusual but practical way. Turbulent flows are chaotic and irregular by nature, however in laminar flows this happens with chaotic advection. We will see that flow patterns can be quite different in these two categories.

5.1 Laminar Mixing

At low Reynolds numbers, the fluctuations are discarded and the flow equations reduce to Navier–Stokes equations. As discussed before (in Chapter 4) materials in convective flows undergo stretching, folding, breakup and coalescence as a result of advection, diffusion and interfacial forces; this is usually referred to as *mixing*. Stretching and folding is caused by advection (i.e. velocity gradient) and breakup and coalescence is a result of the interplay between viscous and interfacial forces. Here mixing will be studied in two parts; mechanical mixing which refers to stretching and folding, (also called stirring in the literature), and mixing caused by interfacial events. Next in mechanical mixing, the interface between different paints will be considered immiscible and the effect of diffusion and surface tension will be neglected. In this section, we will study interesting chaotic patterns some of which are similar to traditional marbling patterns. Then in the following section the flows that include the effect of interfacial events will be studied.

5.1.1 Mechanical Mixing (Stirring)

Mixing, as defined here, is actually a study of flow patterns formed as a result of fluid advection. Mixing can be regular or chaotic. The efficiency of mixing is measured by the amount of stretching and folding; more specifically this is usually the rate of stretching (perimeter length) of a material filament placed in a flow. If this length grows linearly the mixing is regular, if the growth is exponential it is considered to be efficient. If the streamlines are integrable, as in 2D steady flows, the mixing is poor; we can more or less predict the flow patterns according to the streamlines. In laminar flows the streamlines usually look simple; this makes one to expect the advection lines to be as simple and predictable based on the flow vectors. However we observe the complex advection patterns even in fairly viscous marbling flows (see Figure 2.3). This is due to *chaotic advection*. Chaotic advection, as first defined by Aref [2], refers to the chaotic Lagrangian motion of the passive tracers placed in a flow even when the Eulerian solution of the flow is simple.

Mixing is efficient when the Lagrangian motion is chaotic. Chaos has an ambiguous definition as turbulence (see next section): a chaotic system is deterministic, it is sensitive to initial conditions and its long time behaviour is aperiodic and unpredictable. In a chaotic flow the streamlines are nonintegrable. Chaotic mixing is usually studied in dynamical systems. My main objective here is to study chaotic mixing based on these dynamical systems theorems in order to provide tools that can be used by the artists intuitively to generate chaotic advection. From a practical point of view it is a basic process which is done by the traditional marblers intuitively without knowing the underlying dynamics. From the dynamical systems perspective it is not so easy to generate chaos directly from theory. For this reason, certain flow characteristics are obtained usually from or at least by the help of experimental studies. To clarify these concepts, let us first introduce some of these characteristics of chaotic flows.

5.1.1.1 Characterizing Chaos in 2D

In dynamical systems theory, properties of chaotic flows are usually analyzed/quantified by mapping theorems, such as Poincare (horseshoe) maps, lobe dynamics and perturbation theories. These are

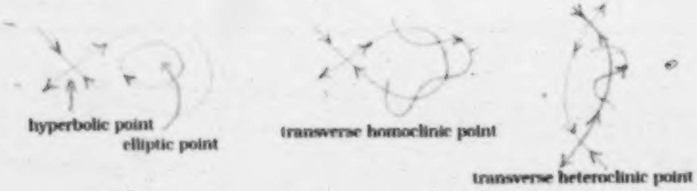


Figure 5.1: Chaotic orbits

beyond the scope of this thesis however without getting into the details of this theory, I will examine these characteristics in terms of the properties of the flow streamlines.

Elliptic and hyperbolic points play an important role in the analysis of chaotic transport in 2D. Particles advected in the flow follow the streamlines and the Lagrangian velocity is integrable, thus stretching and folding can not be efficient. In theory, chaotic transport is defined by the existence of "transverse homoclinic and heteroclinic" orbits. If the stable and unstable manifolds of a hyperbolic cycle join, a homoclinic orbit is formed. If the stable and unstable manifolds of two different hyperbolic cycles join, then it is called a heteroclinic orbit. Stable/unstable manifolds are the regions where the flow joins the hyperbolic point in opposite directions, forward and backward, in time. When the unstable and stable manifolds intersect instead of joining smoothly they are called transverse homoclinic and heteroclinic points (Figure 5.1). (The area formed from the intersection of these stable and unstable manifolds is called a *lobe*. Lobes have a significance in chaos theory due to their properties in characterizing chaos. For example lobe regions are considered invariant as the streamlines can not cross their boundaries. Chaotic advection is measured by the area of the lobes. But these are beyond the scope of this thesis).

If the hyperbolic points can be broken somehow to create transverse heteroclinic and homoclinic points chaos will be produced, or in more practical terms we can say that repeated stretching and folding will give us chaotic advection. This can be achieved by making the flow time dependent by applying certain boundary conditions. In time dependent flows it is possible to cross the streamlines.

To clarify these concepts "blinking vortex" flow [2, 122] can be examined. It is one of the flow configurations designed to study properties of chaotic advection. Blinking vortex flow was introduced by Aref to explain the fluid behaviours under chaotic advection. In this flow two alternatively rotating cylinders are placed next to each other. When both of the cylinders are active, flow is steady and at the hyperbolic point the particle trajectories branch into stable and unstable manifolds smoothly. But when they rotate alternatively these manifolds cross in time and form a complex trajectory. This can only be observed in the Lagrangian motion. A blob placed in this flow is stretched by the first vortex until the first stops and the second starts, the particles trapped in the chaotic region are then caught by the second vortex. In this way, material is stretched and folded repeatedly in the chaotic region with a speed and intensity depending on the periodicity of the rotation of the cylinders. The results can also change depending on the distance between the cylinders and the initial position of the blob. This is a good example to observe how even simple

Eulerian vectors can produce very complex advection trajectories.

Several experiments have been done to study mixing; mixing devices are designed to improve the efficiency of mixing and analyze the behaviour of certain flows. Blinking vortex, journal bearing and cavity flow are some of these. In the well known cavity flow, upper and lower walls of the cavity move periodically creating two vortices inside the cavity; these are two elliptic points with a central hyperbolic point. Journal bearing flow [122] is produced from the rotation of two cylinders. Some of these flows have analytical solutions such as blinking vortex and journal bearing, which makes them easier to study. Up until recently, stirring has been studied in systems where the geometry is fixed. Boyland et.al. [11] were probably the first to apply the modern topology theorems to the stirring problem where the boundaries/stirrers moved. They studied the stirring protocols in case of three stirrers. They used a classification theorem to describe the deformation in terms of isotopy classes. Basically they showed how this theorem is applied to the stirring protocol, and explained the diffeomorphisms in terms of mapping classes. In practical terms, their results proved that if the movement of the stirrers were arranged in a certain way braid-like routes which would for example create an eight shape mixing would be improved. Note that in these topology based studies, the fluid is assumed to be very thick so that it moves in a local region around the boundary and stops when the boundary stops. In this way, only the topological changes can be studied without the interference of other complexities.

5.1.1.2 Experiments

Based on these concepts, I developed some boundary configurations allowing the user to produce chaotic transport efficiently (i.e in an orderly fashion rather than randomly). In all the results in this section, a 2107 cell Voronoi grid is used based on a 500 × 500 liquid grid.

1. In the first example, a basic stretching and folding mechanism by using multiple point boundaries is created. Rotating circle of boundary points are moved by the user to create stretching and folding patterns. Here the position of the boundaries relative to the center are fixed and each point has a velocity tangent to the circle. Intensity of the boundary points is arranged to simulate a smooth circular motion. Viscosity is adjusted low enough to prevent the boundaries from generating small vortices around them, but high enough to let the vectors dissipate and form vortices as a result of the interaction of streamlines. Smoothing level is also adjusted accordingly. This adjustment is made to allow the generation of elliptic and hyperbolic points as a result of overlapping rotations. At each movement rotation is reversed to improve the amount of stretching and folding by crossing the streamlines. In this way one can produce very typical chaotic patterns which are also commonly observed in viscous marbling flows. Figure 5.3a shows a typical marbling design which is probably obtained in a similar chaotic flow. Figure 5.2a-c shows frames from the evolution of the streamlines and Figure 5.3b-g demonstrates the stretch and fold patterns in the advection lines based on a guessed initial configuration. Distance between the boundary centers, speed and duration of each rotation are the parameters affecting the results. A 5th order B-spline interpolation technique is used in advection. Image is represented by four pigments, this is to apply the sharp interface transformation on individual pigments. In case of multiple pigments, density of each pig-

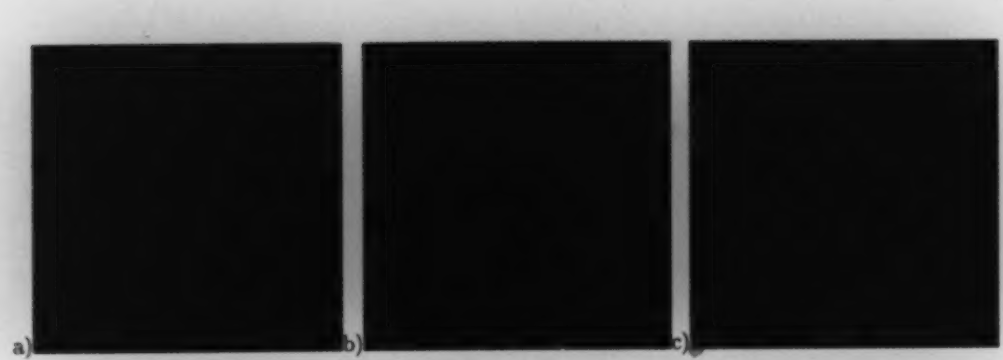


Figure 5.2: Chaotic advection with stretch-and-fold mechanism, a-c) velocity vectors

ment is carried independently, that is advection is applied to each pigment separately. Final color in each pixel is obtained by linear averaging. Although I didn't find it necessary in the current experiments, (since high order B-splines and the sharp interface transformation was sufficient) in case blurring occurs due to linear averaging, high order polynomial color filters can be used to determine the weight of each pigment in a cell by emphasizing the pigment with the highest density.

2. In this example, I used the same type of rotating boundaries but with a smaller radius to create vortices directly from each boundary rotation (Figure 5.4 a,b,c). This is arranged to simulate the chaotic advection produced by interacting vortices. The efficiency of mixing depends on the distance between the vortices, in other words it is the interaction of the vortices rather than their intensity that improves mixing. Figure 5.5d-i shows the expansion of the initial image while it is advected. The reaction parameter value, in Eqn.4.130 is defined as $a = 0.3$. Figure 5.6 shows advection patterns from a similar mixing, and how the material dissolves without the reaction effect.

5.1.2 Interfacial Events

I should first emphasize that, although surface tension plays an important role in several aspects (as discussed in Chapter 2), interfacial events such as breakup and coalescence have not been typically exploited in traditional marbling. At least I haven't encountered such an example. Marbling types which make use of high surface tension effects are mostly based on fixed patterns; paints are dropped on the size and the patterns are obtained from the droplet formation or chemical interaction of droplets without mixing the fluid. In other words, patterns are not formed under convection. Here I'm interested in the effects of topological transitions.

5.1.2.1 Experimental Results

I have performed two experiments; in both of these, fluids are mixed as in chaotic mixing experiments, using rotating boundaries and the flow is laminar. There are two layers of fluid which are subject

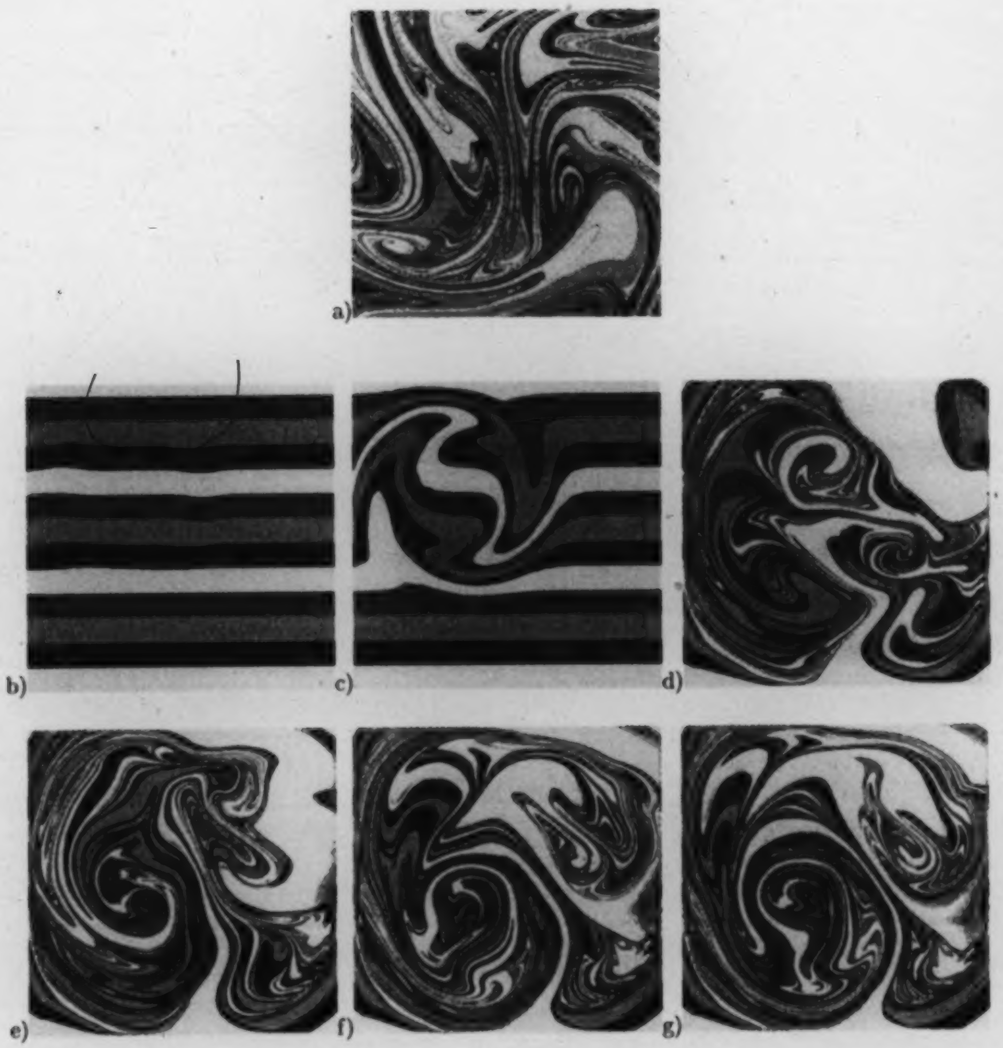


Figure 5.3: Chaotic advection with stretch-and-fold mechanism, with sharp interface advection: a) traditional marbling example, b-g) advection patterns, b) initial configuration guess, c)24 Δt, d)220 Δt, e)310 Δt, f)376 Δt, g)400 Δt

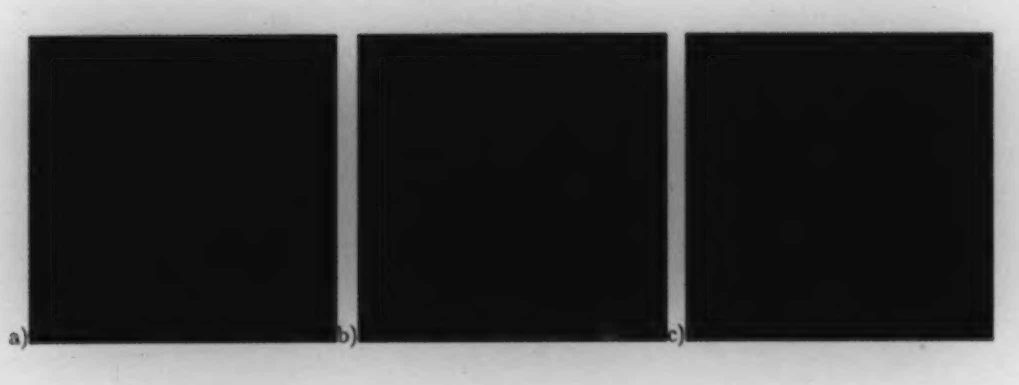


Figure 5.4: Chaotic advection with interaction of small vortices: velocity vectors.

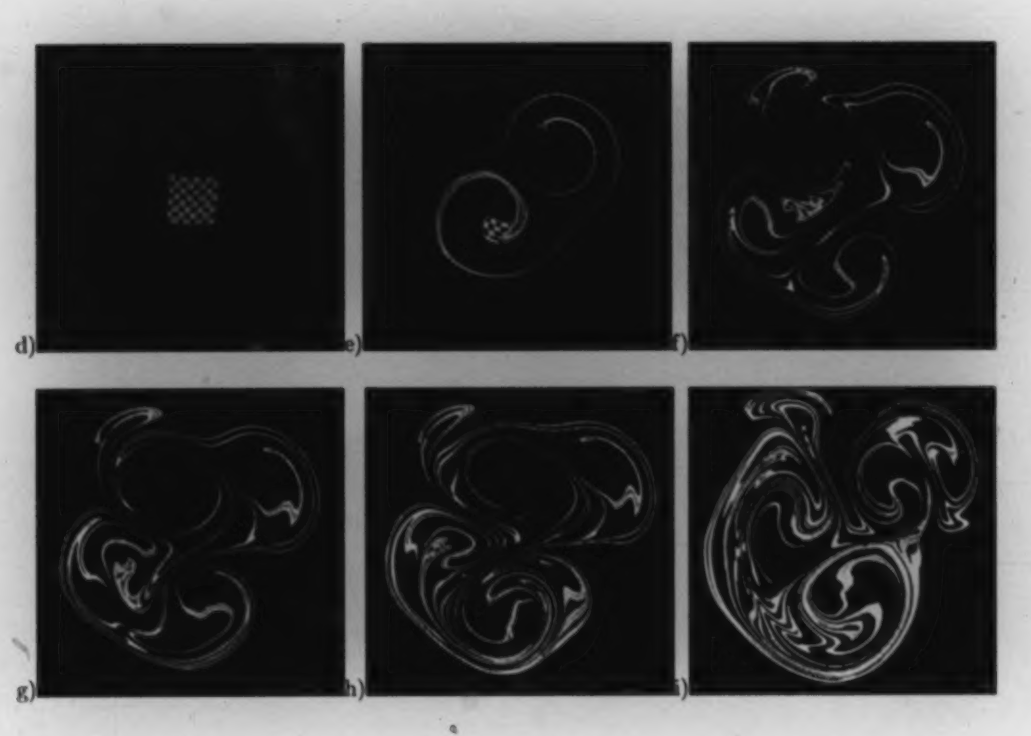


Figure 5.5: Chaotic advection with interaction of small vortices, with chemical reaction.

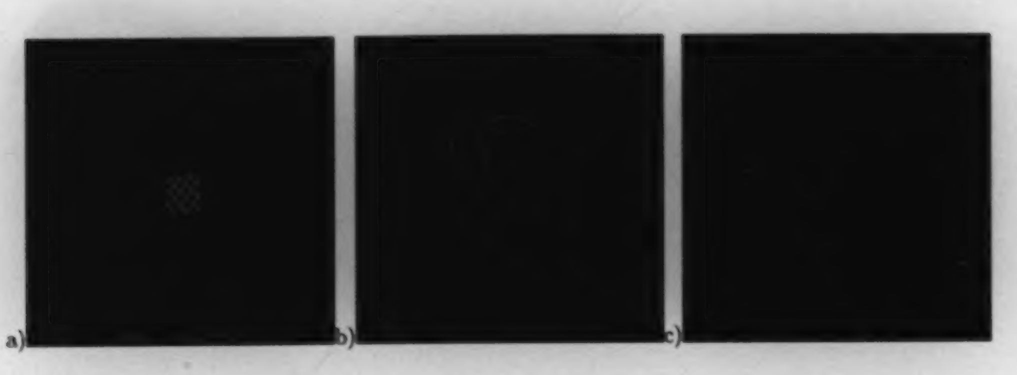


Figure 5.6: Chaotic advection with interaction of small vortices, without chemical reaction.

to the same flow conditions (that is, which are advected with the same flow). The layer on top is affected by high surface tension forces which are described by the Cahn-Hilliard equation while in the background fluid these forces are neglected. In order to increase the scale of topological events and to speed up the diffusion process I applied the *nonlocal effects* method (described in Chapter 4). Based on the properties of this method, its identifying parameters (interface thickness, $\hat{\xi}$ and the magnitude of the nonlocal term $\hat{\alpha}$) can be adjusted in direct proportion. As the interface thickness is increased, the interaction range also increases however the interface becomes more diffusive. To compensate for this the magnitude of the nonlocal term can be increased. This will also increase the droplet survival limit which is determined by the Capillary number. In other words, interface thickness also determines the range above which the droplets survive; droplets smaller than the interaction range tend to disappear. Accordingly, I defined the following parameter values, $\hat{\xi} = 0.002$, $\hat{\alpha} = 0.1$ in these experiments. In both of the experiments the same interface parameters are used, only the initial fluid configurations are designed differently to demonstrate the breakup events in the first and to emphasize the coalescence in the second. (In the first example the top layer is rendered with bump mapping.) In Figures 5.7 and 5.8 we can see the continuous stretching and folding of the background fluids while the droplet dynamics can be observed on the top.

5.2 High Reynolds, Turbulent and Fluctuating Motions in Digital Marbling

In this section turbulent and fluctuating motion in digital marbling is classified based on an analysis of turbulent and high Reynolds number flow behaviours that occur in traditional marbling. High Reynolds flow simulation in digital marbling, though inspired by the traditional technique, is not limited to the conventional flows and covers a wide range of (and not necessarily realistic) flow types for artistic purposes.

I classify the complex motion that occurs in marbling in three categories. The first part constitutes a more realistic simulation of high Reynolds flows, in which the turbulence is mostly triggered by boundaries. This part basically involves the simulation of vortex shedding that occurs at high

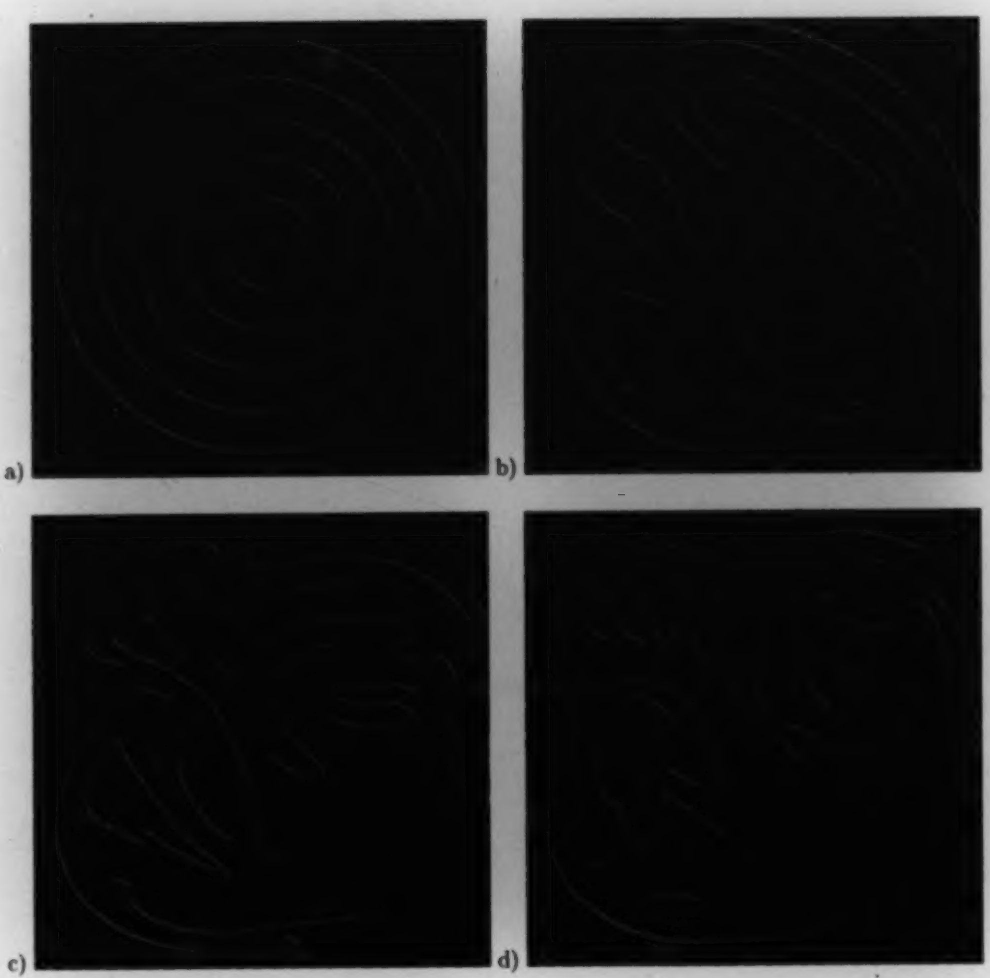


Figure 5.7: Breakup and coalescence, a) $t = 0$, b) $t = 46 \Delta t$, c), $t = 78 \Delta t$, d)$t = 104 \Delta t$.

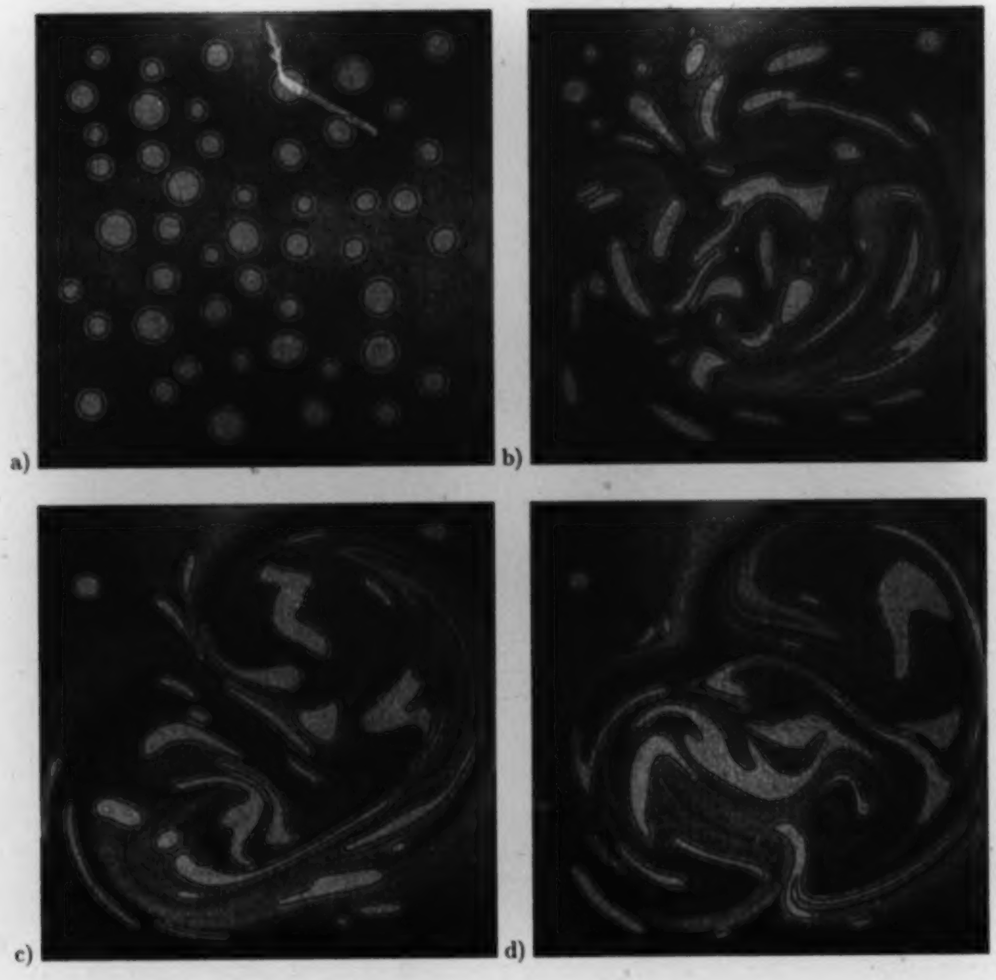


Figure 5.8: Coalescence and breakup, a) $t = 0$, b) $t = 43\,\Delta\, t$, c), $t = 80\,\Delta\, t$, d)$t = 133\,\Delta\, t$.

Reynolds numbers. In the second part, I introduce a statistical model to simulate realistic and non-realistic effects caused by mesoscale fluctuations from small scale oscillations to large scale eddies. In other words, this part involves the simulation of turbulence and complex flow behaviours caused by small scale fluctuations. Finally in the last part, I present a method to model the concentration fluctuations in the advected material. In other words, this part considers the eddy (turbulent) diffusion. But first let us review basic characteristics of turbulence.

5.2.1 What is Turbulence ?

One of the reasons why turbulence is considered one of the most difficult problems in physics is due to lack of a precise definition. Characteristics of turbulence are determined from experimental observations. Two basic laws of turbulence are derived in this way: the first law states that in a turbulent flow, at very high Reynolds numbers, the velocity correlation is proportional to the two-thirds of the distance, $\langle v \rangle \sim r^{2/3}$, and the second one states that the energy dissipation is constant.

Based on observations, turbulence is highly irregular (chaotic), time dependent, rotational and it appears on a wide range of length and time scales. It is sometimes described as the fluid motion formed by eddies with a wide spectrum of sizes. The interaction between eddies at different scales are defined as follows: the largest eddies, also considered the fundamental scale of turbulence, are in the order of the main flow, they appear first and obtain their energy from the main flow, thus they are affected by the boundaries and the changes in the mean flow. In 3D turbulence, energy is transferred from large eddies to smaller eddies, then it is dissipated by the viscosity at the smallest scales. This energy cascade from larger to smaller scales is called vortex stretching, it describes how vortices deform each other and turn into smaller and smaller eddies while increasing in number. In other words, vorticity can be amplified in 3D, thus it is not a conserved quantity. In 2D however, energy dissipation characteristics are different; there is no vortex stretching and vorticity is a conserved quantity. As a result, unlike 3D turbulence, large eddies do not turn into smaller eddies, and separated scales interact with each other. In fact, when viscosity is small energy dissipation is very slow: vortices decay very slowly and in a long time they merge into larger vortices. In addition, in 2D, the fluctuations at large scales are considered important whereas in 3D only the fluctuations at the smallest scales are of concern. Here "large scales" refers to the scales that are comparable to the energy injection scale where grid forcing which is at mesh scale takes place or in other words it indicates the mean velocity scale. This property can be used to justify the application of random forces at different scales in my model. As mentioned before, marbling flow is considered two dimensional due to the chemical and physical properties of the materials involved. I would also like to mention the soap film experiments which are commonly used in two dimensional turbulence studies in CFD [158, 75]. Soap film [25] is a thin layer of fluid created by surfactants with hydrophobic and hydrophilic molecules similarly in marbling fluids (see Chapter 2). Its significance is due to the close approximation of the thin layer of fluid to two dimensional flow. Some of these studies will be referred to during the text.

5.2.2 Vortex Sheet Simulation

In this section, fluid behaviour caused by boundaries at high Reynolds numbers, more specifically vortex sheet dynamics will be considered. It is well known that, (and as we will also see in real experiments), at high Reynolds numbers, when an external object moves through a fluid or as the flow goes past a stationary object, organized vortex structures are produced. Vortex sheet refers to these coherent vortex structures created by sharp boundary edges.

We have seen the flow patterns produced by external objects in laminar flows. As the boundary speed is increased or viscosity is decreased (that is, at higher Reynolds numbers) vortex sheet development starts. Figure 5.21 shows the flow paths (from my experiments) for two different boundary speeds. Depending on the fluid and boundary properties very interesting flow patterns can be created. Vortex sheet patterns are not a typical characteristic of traditional marbling designs, in fact I haven't encountered such a real marbling example. This is probably because certain boundary speed/viscosity and surface geometry configurations are required. Especially when the fluid viscosity is low, flow evolves (and decays) very fast, making it difficult to capture the patterns on the paper.

Vortex sheet patterns which are considered instability patterns that develop downstream the boundary. Above a certain Reynolds number they become turbulent, that is boundary motion creates turbulence however below a certain Reynolds number they are organized, periodic and persistent, that is, they do not decay for a long period. This means, vortex sheet evolution takes place only within a certain range of Reynolds numbers.

Although it is locally unstable, this flow range is considered globally stable. Mathematically vortex sheet patterns are defined as tangential discontinuities along a surface where the vorticity is concentrated. In 3D flow, when vorticity is concentrated in a region this is usually called a *vortex filament*. This flow type has been of interest in CFD studies as it shows some contradictory behaviour considering the conventional understanding of turbulence; this is due to its global stability and periodic, organized characteristics whereas tangential discontinuities are expected to produce turbulence according to common belief. In CFD 2D vortex sheet development has mostly been investigated in soap film experiments.

Although it is not completely understood, initial creation of these vorticity structures is described by the boundary layer theory. Basically fluid molecules sticking to the object's surface produce a velocity difference between the object's surrounding and the main stream flow. As a result, a thin layer of fluid, which is called the boundary layer, is formed near the surface. At low Reynolds numbers, velocity linearly varies from the object's velocity to the main stream velocity. On the other hand, at high Reynolds numbers velocity in this region becomes turbulent (*turbulent boundary layer*) and the pressure difference due to high velocity gradients cause the boundary layer separate from the object's surface, creating vortices which form periodic patterns as they are carried with the flow. These vortices eventually decay due to viscous forces. Structure of these patterns mostly depend on the surface geometry.

5.2.2.1 Numerical Modeling

Numerical modeling of this flow is obviously difficult as the flow evolution from the boundary layer is not well defined. Aside from that, direct numerical simulation of vortex shedding by solving Navier–

Stokes equations is a challenging task as it requires high resolution and accuracy in discretization. One of the most popular cases studied in CFD is the "flow behind a cylinder", where the vortex sheet that is produced behind a cylinder that is placed in a uniform flow is considered. In solving this problem, and similarly in other cases, usually an adaptive grid is used which is refined around the boundary and the vorticity regions [9, 94, 96]. Generally finite element or high order finite volume methods are used for spatial discretization and implicit methods for time discretization to maintain stability and also to decrease the effect of numerical diffusion. Implicit formulation, also considering the incompressibility condition, becomes more difficult when high order accuracy is required on a complex grid structure. Furthermore when the boundary is moving, adaptive refinement technique becomes computationally inefficient and impractical especially for an application like marbling.

Concentrated vorticity evolution of vortex sheets have been more commonly modeled by using vorticity equations. Vortex sheet is a special case of vorticity dynamics. General vorticity formulation of Navier–Stokes in 3D is given by,

$$\frac{D\boldsymbol{\omega}}{Dt} = (\boldsymbol{\omega} \cdot \nabla)\mathbf{u} + \nu\nabla^2\boldsymbol{\omega}, \tag{5.2}$$

where $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ represents the vorticity. In 2D, vorticity is perpendicular to velocity so the first term on the right hand side of Eqn. 5.2 vanishes (which also implies the absence of vortex stretching) and the vorticity equation reduces to:

$$\frac{D\omega}{Dt} = \nu\nabla^2\omega. \tag{5.3}$$

In the Eulerian description, in order to enforce the incompressibility constraint a scalar stream function term is introduced, according to which the velocity is defined as follows:

$$u = \frac{\partial\psi}{\partial y}, \quad v = -\frac{\partial\psi}{\partial x}. \tag{5.4}$$

The stream function vorticity relation can then be described by the Poisson equation which is obtained by taking the curl of the velocity which is represented in terms of the streamfunction:

$$\omega = -\nabla^2\psi. \tag{5.5}$$

In 2D, since vorticity is represented by a scalar field, vorticity formulation reduces to scalar advection-diffusion problem, accordingly the discretization is simplified. In this case, spectral methods become one of the alternative solutions. Spectral methods may produce less numerical dissipation hence they are preferable in rectangular domains as long as the external boundary implementation is not required. Let's recall that, since the spectral discretization is based on a global transformation it does not allow the imposition of external local boundary conditions. It should be mentioned that defining the boundary conditions for the vorticity and streamfunction is also problematic in finite difference discretization. Numerical studies for the vorticity equation are numerous in CFD. Here only some of these which are related to vortex sheet modeling will be mentioned. For example, Couder et.al.'s work [26] is significant as it is one of the earliest studies in which the vortex sheet development is investigated experimentally on soap films. In this study vorticity equation is solved with a spectral method and the numerical results are compared to the real experiments. They however had to use a super dissipative term to control the dissipation effect of the viscous term by limiting it to smaller scales. Another drawback of this method is its limited resolution. In graphics, in [166], Yaeger et.al. used a similar pseudospectral method but without the super diffusive effect to simulate the rotating motion in planetary atmospheric flows. In simulation of strong zonal flows the terms responsible for Kelvin-Hemholtz instabilities which are the main source of vortex sheet generation have been discarded. I should also point out that none of these methods based on the spectral discretization incorporates the external boundaries in the fluid model; initial formation of vorticity is not modeled directly but given as an initial condition. For the vortex sheet simulation conventionally inviscid (Euler) equations have been used. In inviscid formulation of the fluid equations diffusive term is excluded:

$$\frac{D\omega}{Dt} = 0. \tag{5.6}$$

In Eulerian discretization since the stabilizing effect of viscous forces is absent computation is susceptible to instabilities. Particularly in vortex sheet simulation singularity formation due to surface instability of the vortex sheet constitutes a problem. Mainly because of its natural adaptability to complex surfaces, Lagrangian discretizations of the inviscid equation, that is, so called vortex methods, have been a more popular choice for modeling vorticity dynamics. In vortex methods, vorticity is represented in terms of vortex particles. Particle velocities are obtained from the Biot-Savart law which represents the solution of the Poisson equation in the Lagrangian frame. Vorticity of each particle (in vortex methods) is represented as a sum of N point vortices as follows:

$$\omega(\mathbf{x}) = \sum_{i=1}^{N} \Gamma_i \delta(\mathbf{x} - \mathbf{x}_i), \tag{5.7}$$

where δ denotes a delta function and Γ_i represent the strength of the point vortices located at $\mathbf{x}_i$. Consequently each particle's velocity can be represented by:

$$u_i = \sum_{j=1, j\neq i}^{N} \Gamma_j K(\mathbf{x}_i - \mathbf{x}_j). \tag{5.8}$$

This is called the Biot-Savart integral [130] where the kernel is defined as:

$$K = \frac{1}{2\pi \left|\mathbf{x}\right|^2}(-y, x). \tag{5.9}$$

In the original vortex method (which is called the point vortex method in [129]) this kernel was used. This method did not produce smooth results and was susceptible to singularities. Later Chorin introduced the vortex blob method [22] where a smoothed kernel (such as a localized Gaussian function) was used for regularization purposes. This particle simulation constitutes an N-body problem which becomes computationally very expensive as the grid resolution (represented by the number of particles) increases. In order to ease the N-body computation and also to take advantage of the adaptability of the Lagrangian particle integration while making use of the fast Poisson solvers at the same time, vortex-in-cell methods have been proposed [23] (vortex-in-cell method is also suitable for incorporating diffusion into the vortex method integration). In this method, particles move on a fixed grid, Poisson equation for the streamfunction from which the velocity of the particles are obtained, is solved on this fixed grid. Particle motion is performed in the Lagrangian frame. Velocity values are transferred from the grid to the particles and vice versa via interpolation. This method has been used in graphics [62] to model vortex motion. However as well as being prone to discontinuities due to particle representation of the velocity, vortex-in-cell method also suffers from the information loss caused by the interpolation which prevents the modeling of fine vorticity structures.

At this point, let's look into the special case of vortex sheet motion. Vortex sheet indicates a line with infinite Reynolds number where vorticity is concentrated and tangential velocity has a discontinuity. In this region flow is very sensitive to small perturbations which develop as instabilities. Since Biot-Savart equation is not integrable on the vortex sheet due to singularities, an integration based on the average tangential velocity along the width of the vortex sheet has been introduced by Birkhoff and Rott as follows:

$$u(\Gamma, t) = pv. \int \Gamma(x', t) K(x', t) dx'. \tag{5.10}$$

Birkhoff-Rott equation, [130], as represented in the continuous form here, is a special case of Biot Savart integral; it is the vortex sheet formulation in the Lagrangian frame. In this equation, the jump in the tangential velocity identifies the strength of the vortex sheet, Γ and *pv* stands for the Cauchy principle value integration which basically refers to the integration taken as the limit goes to the singularity which is at the vortex sheet in this case. This formulation shows how the vortex sheet problem is reduced to an initial value problem. In other words, it explains the evolution of the vortex sheet after the boundary layer has created the initial shape of the perturbation.

Discretization of this integration is also considered an ill-posed problem due to the singularity which causes the Laplacian operator to become ill-posed beyond a critical value. Proof of this theory is beyond the scope of this thesis [130] however in simplistic terms this pertains to the fact that the sheet has an infinite Reynolds number and the spiral has an infinite roll-up. As a remedy to this problem in numerical implementation regularization methods have been proposed. Krasny [99, 100] has applied the same regularization idea of the vortex blob method to the vortex sheet equation:

$$\frac{dx_j}{dt} = \sum_{j=1, j \neq i}^{N} \Gamma_j K_\delta(\mathbf{x}_i - \mathbf{x}_j), \tag{5.11}$$

where the kernel is defined as:

$$K_\delta(\mathbf{x}) = K(\mathbf{x}) \frac{|\mathbf{x}|^2}{|\mathbf{x}|^2 + \delta^2}. \tag{5.12}$$

In this formulation as $\delta \to 0$, that is, as the vortex sheet singularity is approached, the integration approximates the Cauchy principle value integral.

In vortex sheet simulation Lagrangian vortex methods generally suffer from the discontinuity caused by representing a continuous velocity field by discrete particles. This discontinuity being very sensitive to vortex sheet singularities obviously creates instabilities in the particle motion and results in poor convergence. While being subject to these difficulties, Krasny's model [99, 100] provided high resolution idealized vortex sheet evolution. They achieved convergence at a considerably high grid resolution by adjusting the smoothing parameter. (Note that the regularized "vortex blob" method is considered an idealized solution since its relation to the real viscous flow equations has not been fully explained in theory.) However this is still a computationally inefficient method considering the implementation difficulties in the N-body problem (such as tracking the particles and deleting and inserting new particles according to the vorticity evolution) and the computational time requirement.

5.2.2.2 Level Set Vortex Sheet Model

For the vortex sheet problem in marbling, I propose a level set based Eulerian approach. Level set idea for vortex sheet modeling was introduced by [70] in CFD. Note that this is not a common approach in CFD; in fact this was the only study I have encountered in literature where level set is used for modeling surface instabilities. Based on the original theory given in [70], by providing a suitable discretization and improvising the level set control, I will use this method for realistic vortex sheet and turbulence modeling as well as a tool for other marbling related artistic effects. First let's review the basics of the theory. In this method a vortex sheet is represented by a level set function, as the zero level set of a signed distance function, and its evolution in the Eulerian frame is described by the vorticity streamfunction equations (2D Eqns 5.4, 5.3, 5.5). As a remedy for the singularity problem a smoothing function is applied to the vorticity parameter. The main idea is to decompose vorticity into a smoothed level set function, $F(\varphi)$, and a strength parameter, Γ, as follows:

$$\omega = F(\varphi)\Gamma, \tag{5.13}$$

where φ denotes the level set parameter and F indicates a smoothing function. When this representation of the vorticity is inserted in the vorticity equation the following set of equations are derived,

$$\frac{\partial \varphi}{\partial t} = -\mathbf{u}\cdot\nabla\varphi, \tag{5.14}$$

$$\frac{\partial \Gamma}{\partial t} = -\mathbf{u}\cdot\nabla\Gamma, \tag{5.15}$$

$$\mathbf{F}(\varphi)\mathbf{\Gamma} = \nabla \times \mathbf{u}. \tag{5.16}$$

Since in 2D the vortex sheet doesn't change sign the vortex sheet strength can be normalized, that is assumed constant, $\Gamma = 1$. As a result, the second equation (Eqn.5.15) can be discarded and the equation set reduces to the advection of the level set parameter and the Poisson equation for the streamfunction:

$$\frac{\partial \varphi}{\partial t} = -\mathbf{u}\cdot\nabla\varphi, \tag{5.17}$$

$$F(\varphi) = -\nabla^2\psi. \tag{5.18}$$

In this way, a smoothed version of the level set data is applied to the Poisson equation instead of the level set itself in which case the Poisson solution would not be convergent. This simple but elegant idea based on smoothing the level set function constitutes the core of this method. Moreover the level set handles the topological transitions, such as the merging of vortices automatically. Note that, this smoothing approach can also be compared [70] to the regularization method proposed by Krasny for the Lagrangian equations [99, 100].

5.2.2.2.1 Discretization

Vortex sheet equations are discretized on a rectangular grid with the same resolution as the image grid to obtain higher resolution in vorticity evolution. Efficiency of the numerical discretization thus needs to be considered. In the original method [70] an ENO method is used for the advection equation for its adaptability to shocks considering the singularity of the vortex sheet level set. And a second order finite difference elliptic solver is used for the Poisson equation. Lower order methods are chosen to make use of the smoothing effect of the numerical truncation error.

I used a semi-Lagrangian method for the level set advection. Considering its limited stability range, ENO approximation would not be practical for this simulation. In order to preserve the stability of the level set and to be able to follow the level set evolution accurately (while preserving the topological transitions) relatively small time steps are taken. However still the semi-Lagrangian approach was much more efficient than the ENO solution. For the spatial approximation, I used bilinear interpolation instead of high order B-splines to take advantage of the numerical smoothing and to prevent possible instabilities at the vortex sheet evolution (which actually were evident in my experiments with higher order methods). For the Poisson equation, I used a FFT based spectral method using Matlab's "discrete sine transform" routines based on the following algorithm:

```
Poisson: $F = -\nabla^2\psi$
1. $\mathbf{Z} = DST(DST(\mathbf{F})')$;
2. $Z_{ij} = \frac{Z_{ij}}{f_i+f_j}$;
3. $\psi = DST(DST(\mathbf{Z})')$
```

where DST stands for the "discrete sine transform" and $\hat{f}$ denotes the Fourier coefficients. This method is chosen mainly for its computational efficiency; for example on a 1000 × 1000 grid, Poisson solution using a matrix decomposition method (such as the iterative methods described in Chapter 4) would take hours while with the FFT method the computation time was only a fraction of a second. This spectral method also has a very high order of accuracy; this improves the spiral roll-up computation in the vortex sheet, in other words, it provides a relatively high resolution vortex sheet discretization. Discrete sine transform is chosen for its boundary properties. It is based on Drichlet boundary conditions, whereas Fourier transform assumes periodic boundary conditions. In order to simulate the velocity interaction at the walls more realistically periodic boundary conditions are not considered. Note that this spectral solution requires a uniform rectangular grid.

The level set method based on this discretization is very practical both in terms of computation and implementation. It can easily be applied to high resolution grids (such as 1000 × 1000) within considerably low computation times. In addition, Eulerian formulation makes the implementation very easy; one doesn't have to deal with organizing particles to adjust them to the vortex sheet resolution or with controlling the instabilities (i.e. the irregular motion that occur at vortex roll-ups). The accuracy of this method regarding its convergence properties as the vortex sheet strength

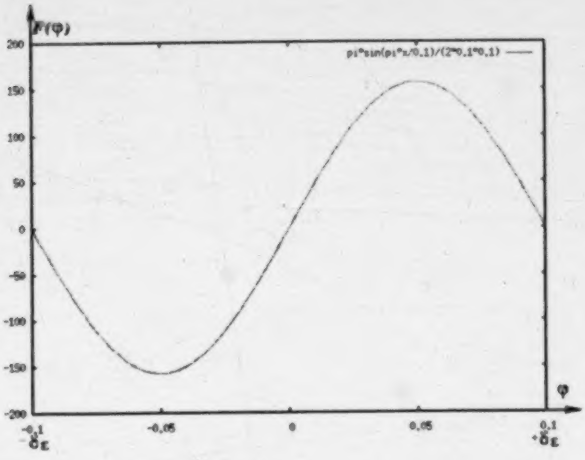


Figure 5.9: Level set transition with the sinusoidal smoothing.

goes to zero, or its approximation of the infinite spiral roll-up (which are considered some of the main criteria in numerical vortex sheet computation) can be disputable. However I found this level of accuracy sufficient for the marbling application.

5.2.2.2.2 User Control

Having defined the discrete model, I will briefly describe the properties of the level set initialization, smoothing function and the basic user control. First, let's emphasize that the initial level set data corresponding to the boundary paths needs to be predefined by the user. Currently, each boundary motion is taken as a whole, in other words, the flow starts only when the level set line (describing the boundary path) is complete. This actually is not entirely unrealistic particularly when the boundary speed is high and the initial perturbation starts to shape slightly after the boundary is released. However with the current system, it won't be difficult to develop a more realistic user interaction in the future where the level set data is updated in real time as the user draws the path on the screen.

The level set data is defined as a signed distance function where the zero level set identifies the velocity circulation, that is it represents the level around which the circulation is completed. Smoothing function needs to be designed according to the level set properties. For example when the level set, φ, changes between $[-1, 1]$ linearly, a sinusoidal smoothing function can be defined as [70]:

$$F(\varphi) = \begin{cases} \frac{\pi \sin(\frac{\pi\varphi}{\delta_\varepsilon})}{2\delta_\varepsilon^2} & |\varphi| < \delta_\varepsilon \\ 0 & \text{elsewhere} \end{cases}, \tag{5.19}$$

which will give the transition shape for the initial data as shown in Figure 5.9. In equation 5.19, δ_ε indicates the width of the vortex sheet.

In general, the basic control parameters defining the vortex sheet evolution and the flow properties can be classified as follows: smoothness and sign transition of the initial level set data, shape

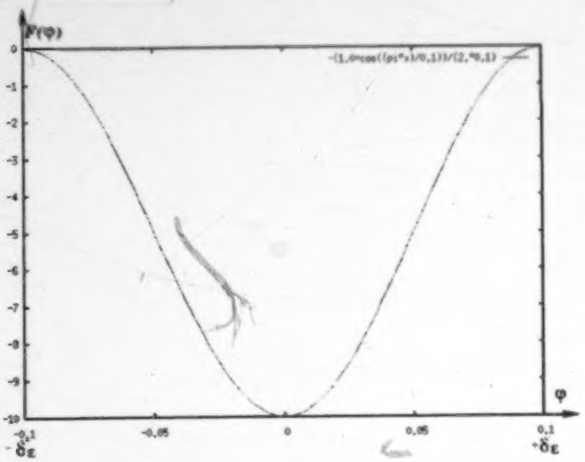


Figure 5.10: Level set transition with the cosine smoothing.

of the transformation function and the vortex sheet width and the advection step size.

5.2.2.2.3 Experimental Results With the level set vortex sheet model, based on the given control parameters, one can design various vorticity based flows as well as other artistic effects including traditional *ebru* designs. The flow types will be classified according to the characteristics of different parameters and their relations as follows.

1. Basic Initial Data Properties

Let us take a sinusoidal initial perturbation defined by the level set function:

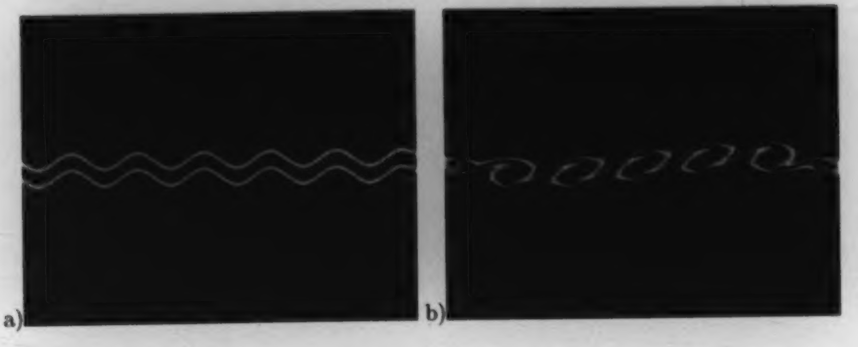
$$\varphi(\mathbf{x}) = x + A\sin(f\pi y), \tag{5.20}$$

where A represents a scaling factor, and f denotes the frequency. This initial shape produces Kelvin-Helmholtz instability if the following smoothing function is applied:

$$F(\varphi) = \begin{cases} \frac{-(1+\cos(\frac{\pi\varphi}{\delta_\varepsilon}))}{2\delta_\varepsilon} & |\varphi| < \delta_\varepsilon \\ 0 & \text{elsewhere} \end{cases}, \tag{5.21}$$

with the parameter values, $A = 0.05$, $f = 6$, $\delta_\varepsilon = 0.1$ and $(x, y)\epsilon[-1, 1]$ on a 600×600 grid. Figure 5.10 shows the transition shape for this function and Figure 5.11 shows the transformed level set values visualized in Matlab color plot. We see how this function creates two parallel zero level sets (shown in green) which evolve into Kelvin-Helmholtz instability.

If the sinusoidal smoothing function is used, the same data initiates the Karman vortex sheet development induced by a flat plate. Figures ??a-c show the evolution of the transformed level set data with zero level set at the center and asymmetric sinusoidal shape on the sides. Figures ??d-f show the Karman vortex street evolution in advection patterns.



a) b)

Figure 5.11: Level set initialization and evolution for Kelvin-Helmholtz instability.

For the same sinusoidal input a different circulation is obtained when the cosine transformation is shifted as follows:

$$F(\varphi) = \begin{cases} \frac{-(\cos(\frac{\pi\varphi}{\delta_\varepsilon}))}{2\delta_\varepsilon} & |\varphi| < \delta_\varepsilon \\ 0 & \text{elsewhere} \end{cases} . \tag{5.22}$$

In this case the vorticity circulation is completed around the grid. This produces the lengthwise symmetrical deformation as shown in Figure 5.13.

Frequency and amplitude of the initial sinusoidal perturbation affect the evolution and shape of the vortex sheet (i.e they determine the instability types.) For example, when the sinusoidal data is slightly perturbed in the vertical direction as well (as shown in Eqn. 5.23), the vortices tend to separate from the vortex sheet area in one direction (Figure 5.14).

$$\varphi(\mathbf{x}) = x + 0.1\sin(2\pi(y + 0.2\sin(2\pi x))) \tag{5.23}$$

Evidently, slight modifications in the perturbation or the width of the vortex sheet, δ_ε, can change the vortex sheet structure. Note that, these sinusoidal initializations have been investigated in [70, 100] to study vortex sheets created by a flat plate or a cylinder. The initial perturbation shape is identified with the geometric configuration of the boundary. For example if the width of the flat plate is large enough to separate the vorticity evolution at the two ends of the plate, the following transformation function can be used to describe the level set as given in [70]:

$$F(\varphi) = \begin{cases} \frac{-(1+\cos(\frac{\pi\varphi}{\delta_\varepsilon}))}{2\delta_\varepsilon} & -2\delta \le \varphi \le 0 \\ \frac{(1+\cos(\frac{\pi\varphi}{\delta_\varepsilon}))}{2\delta_\varepsilon} & 0 \le \varphi \le 2\delta_\varepsilon . \\ 0 & \text{elsewhere} \end{cases} \tag{5.24}$$

Figure 5.15 shows the evolution of the transformed level set and 5.16 shows the advected image for the same sinusoidal initial data as in Eqn. 5.20 (on a 800 × 800 grid)

2. Secondary Vortices

The advection step in the previous example is adjusted by slightly increasing the time interval to create secondary instabilities. Secondary vortices that are observed in real experiments have been a

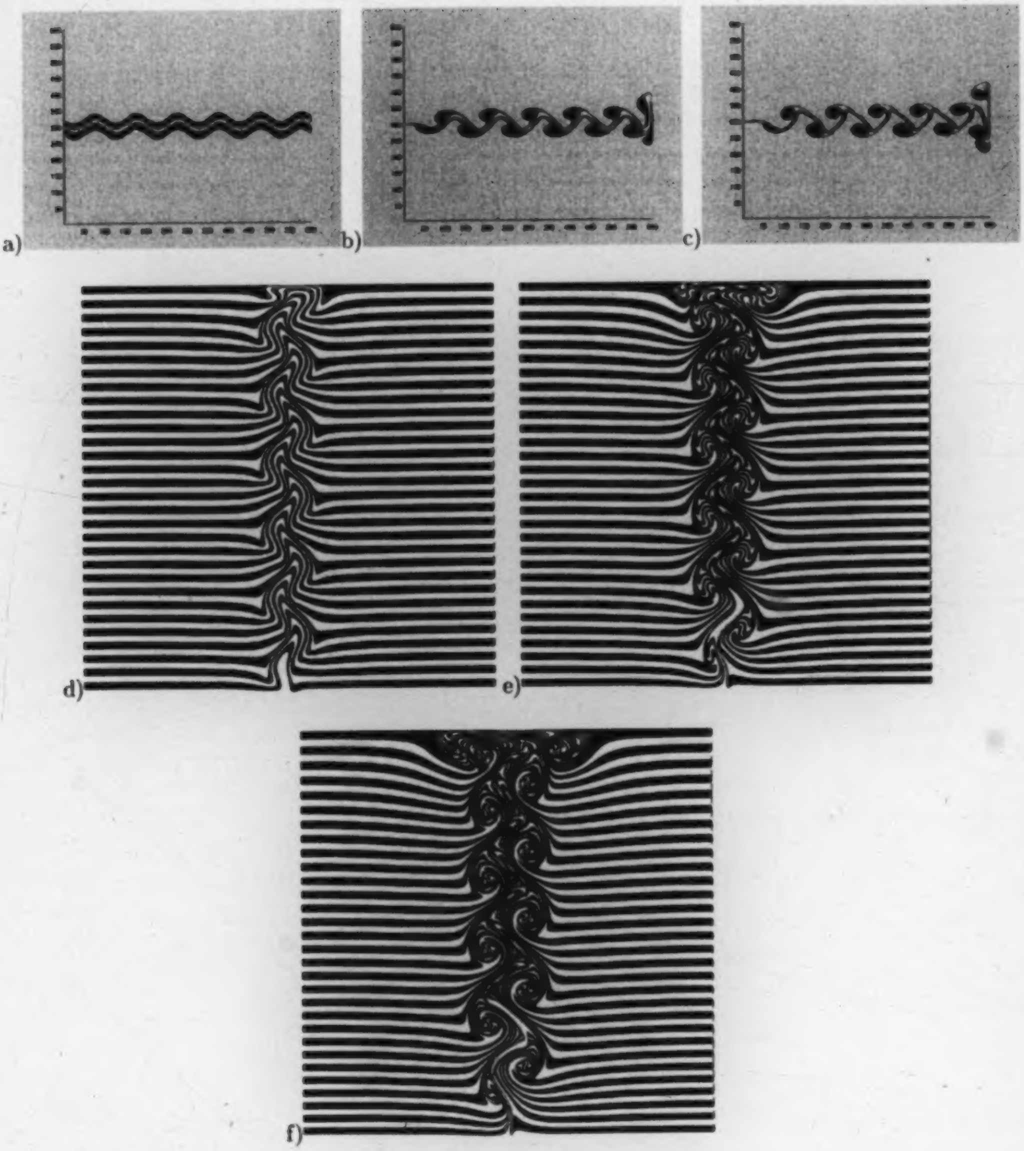


Figure 5.12: **Karman vortex street evolution, a-c) smoothed level set data, $F(\varphi)$, d-f) advection patterns.**

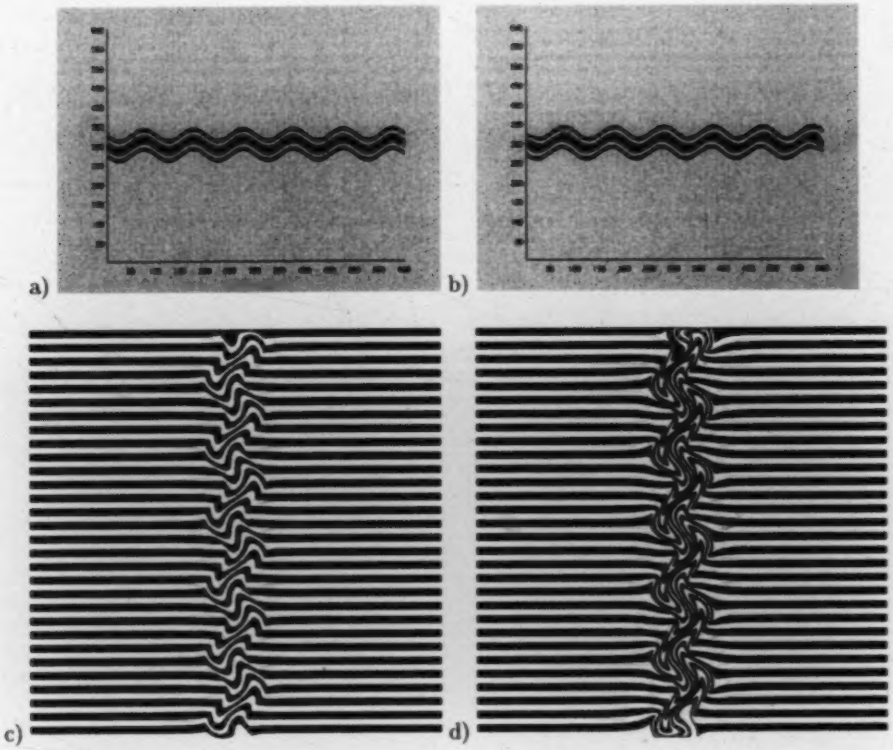


Figure 5.13: Level set, $F(\varphi)$, evolution (a,b) in case of cosine smoothing and advection patterns (c,d).

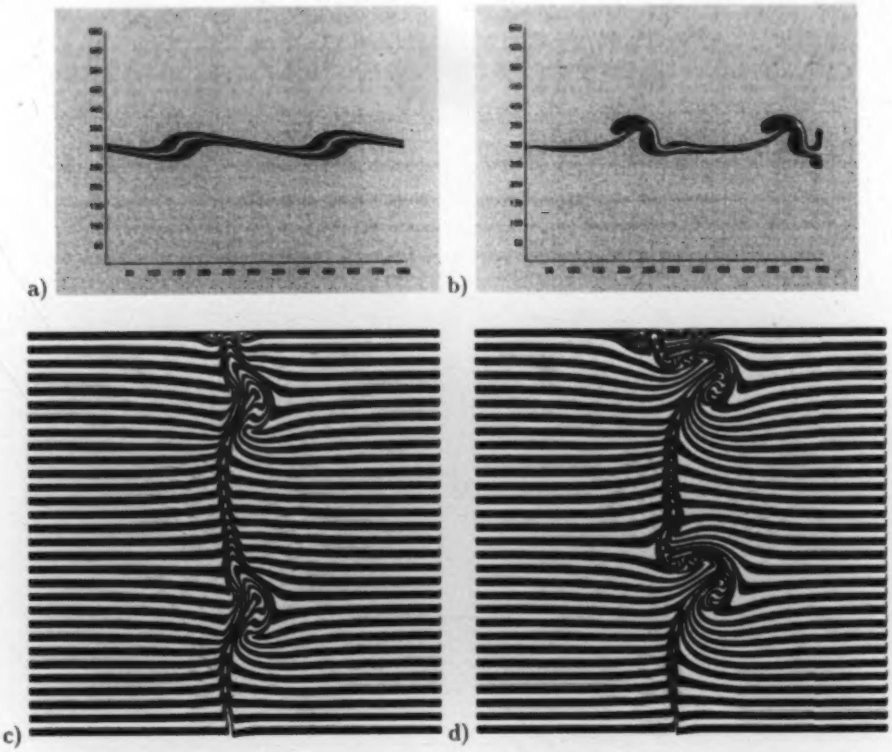


Figure 5.14: **Karman vortex street evolution for sinusoidal input with vertical perturbation. a,b) level set data , c,d) advection patterns.**

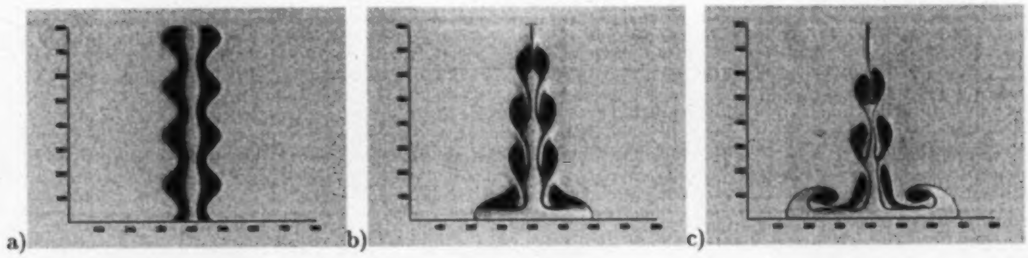


Figure 5.15: **Karman vortex street evolution produced by a flat plate with increased width: level set. a)$t = 0$, b) $t = 13\,\Delta\, t$, c) $t = 26\,\Delta\, t$.**

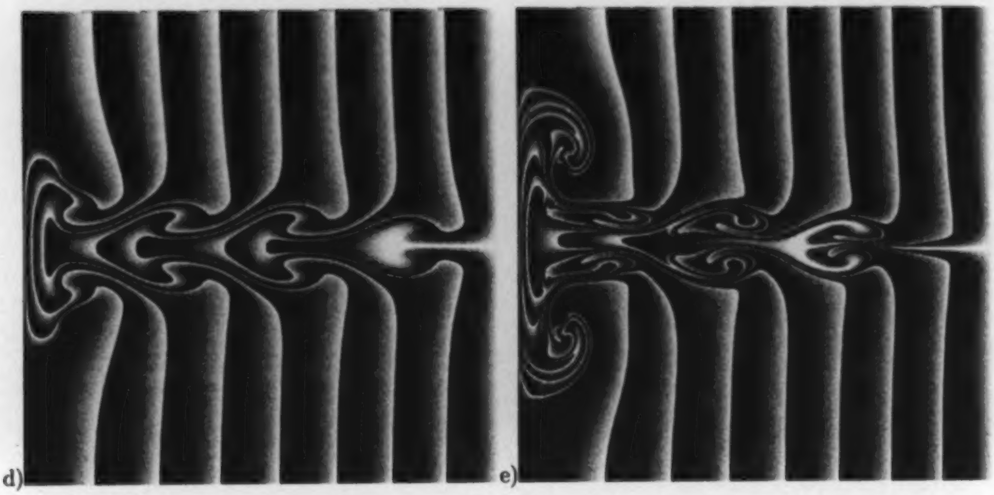


Figure 5.16: Karman vortex street evolution produced by a flat plate with increased width. d) $t = 13\,\Delta\,t$, e) $t = 26\,\Delta\,t$.

subject of interest in CFD and math studies as what causes them or their mathematical properties are not completely understood. There are different theories regarding the origin of the secondary vortices. According to some they are caused by small irregularities in the boundary geometry. On the other hand, according to a stronger argument which is more commonly accepted, they are a property of the flow. The second argument has been supported by several numerical experiments (see the discussion in [1]). In accordance with this theory, secondary vortices are usually modeled by adding small perturbations to the vortex sheet during its evolution [1]. I also experimented with this technique, however later I found a more practical solution based on creating instabilities simply by adjusting the advection step and the smoothness of the level set.

In Figure 5.17 we see how the level set gets thinner as it is advected but it also stays strong (with a high peak in transition) consequently it doesn't dissolve out, instead it continually creates secondary vortices. As the Poisson solution is convergent the system can be considered numerically stable. Figure 5.18 shows the fast evolution of the advected image. The gray level image is chosen to visualize small scale vortices better. Without the secondary instability effect the vortices would continually roll-up in the original vortex sheet shape to form infinite spirals.

Next I experimented with the shape of the smoothing function. The same sinusoidal initialization with the vertical perturbation is used as in Eqn. 5.23. However now the slope of the level set data is smaller so that as a result of sinusoidal transformation softer waves are produced compared to the high peaks in the previous example. How this transition is obtained will be described in the next experiment. When the advection step is adjusted accordingly, again, small secondary vortices are produced, however this time they are perfectly smooth and circular and they evolve into a continuous roll-up. Due to its smoothness the level set data dissolves as it is advected and the flow slows down eventually. Figure 5.19 shows the smooth level set evolution, Figure 5.20 shows the

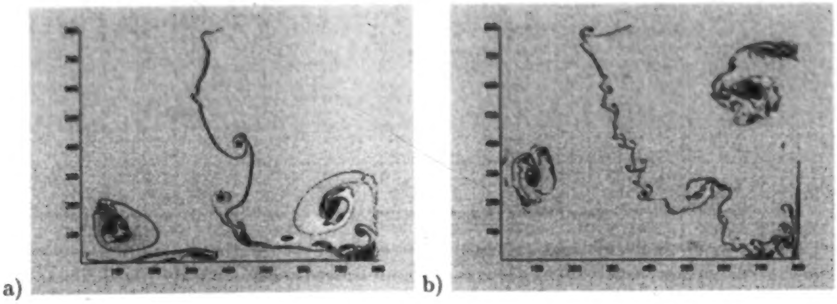


Figure 5.17: Level set evolution in the flow initialized by the flat plate (continuation of the experiment in Figure 5.15). a) $t = 83 \triangle t$, b) $t = 189 \triangle t$

secondary vortices created in this way in the advected image.

3. Comparisons With Real Experiments

I also compared the numerical results from the level set technique with the results from the real experiments. In the real experiments I used a stylus with a flat shape. In the first experiment, the fluids have a low viscosity; only a small amount of dispersant is used, enough to make the colors float. The width of the stylus is small so that it creates typical Karman vortex streets. Figure 5.21 shows the Karman vortex street evolution for different speeds of the stylus. In the numerical solution I altered the amplitude of the sine wave to simulate different speeds of the stylus.

In the second experiment the fluid is more viscous and vortex streets don't develop in full vortices in most cases. Figure 5.22 shows frames from the fluid motion after a few strokes in different directions. In the numerical solution, in order to obtain similar flow patterns I adjusted the frequency and amplitude of initial paths. Note that, the initial configuration is not the same as the original but only an approximation, firstly since the camera motion in the original recording was not stable and secondly due to the resolution adjustment from the video frame to a square image.

In order to obtain a signed distance approximation for the truncated paths, which do not go all the way to the grid edges, I defined the following formula:

```
Signed distance approximation of a truncated line:
1. Take the distance map of the line
2. Take the distance map of the same line which is shifted by a distance.
3. Take the inverse of the image from the second step
4. Add two images from steps 1 and 3.
```

The shift distance determines the cut-off range of the level set (i.e vortex sheet width), it can also be used to control the smoothness. Note that, there are actually more precise signed distance mapping algorithms [167] however they are computationally expensive, and not suitable for this application. Figure 5.23 shows the level set data and Figure 5.24 shows the advected images for 5 different strokes that are drawn similarly to the real experiment.

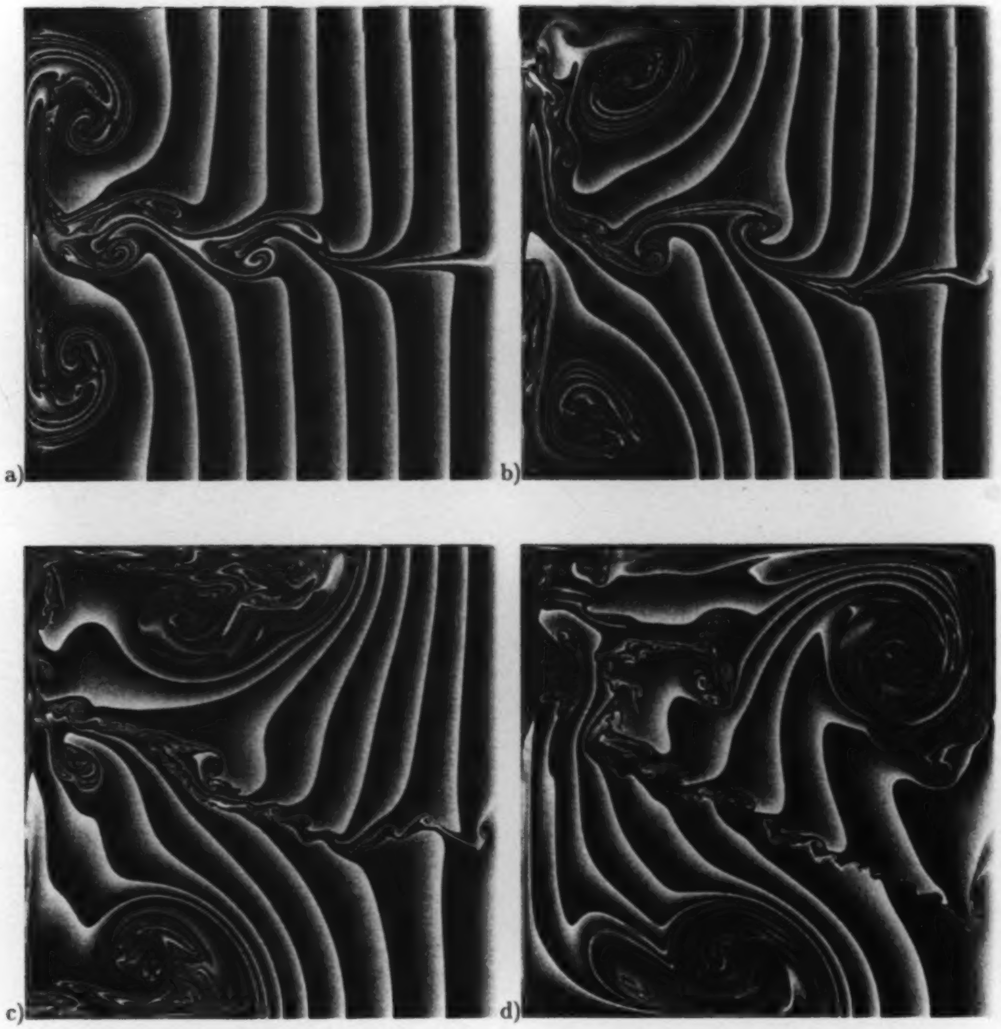


Figure 5.18: Secondary vortex formation in the flow initialized by the flat plate (continuation of the experiment in Figure 5.16). a) $t = 47 \triangle t$, b) $t = 83 \triangle t$, c) $t = 136 \triangle t$, d) $t = 260 \triangle t$.

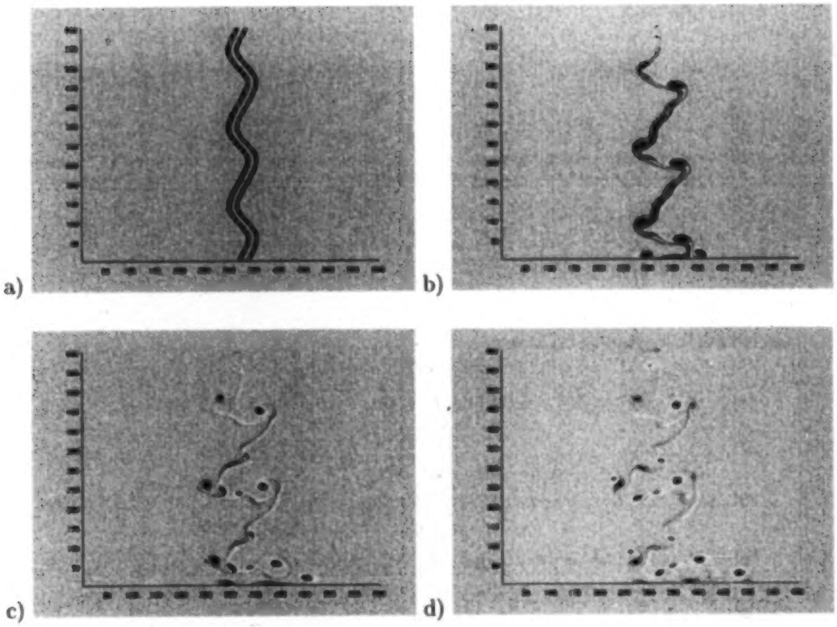


Figure 5.19: **Flat Plate secondary vortex formation: level set data. a) $t = 0$, b) $t = 50 \triangle t$, c) $t = 100 \triangle t$, d) $t = 150 \triangle t$.**

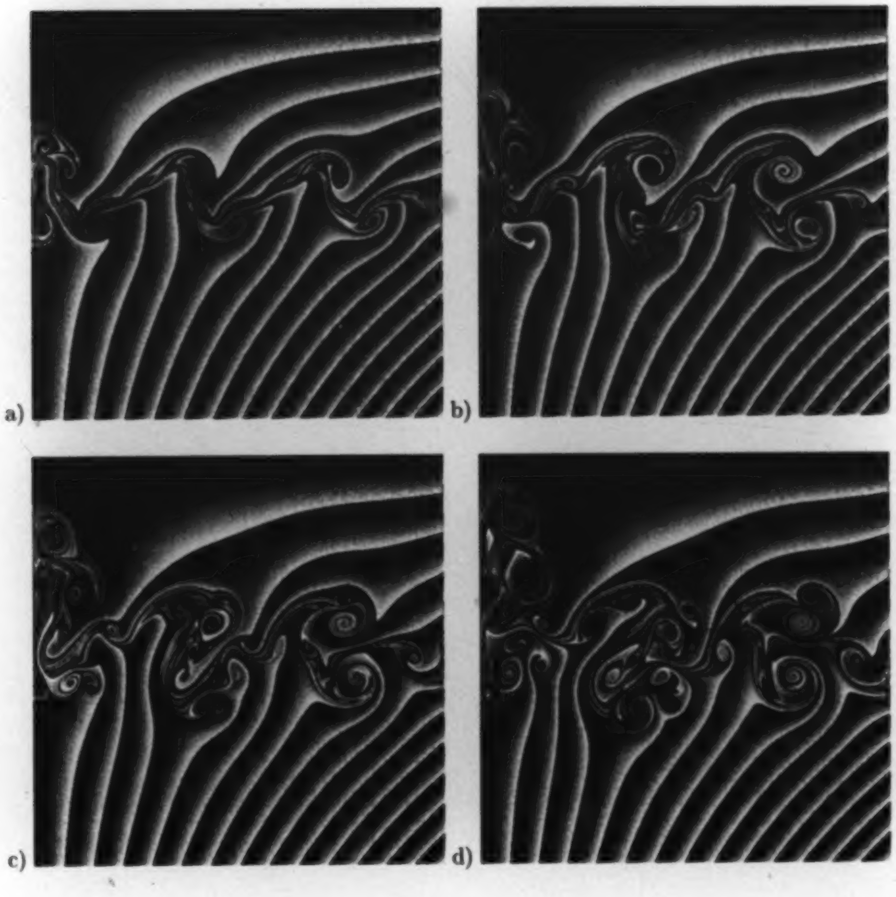


Figure 5.20: Flat Plate secondary vortex formation: advection patterns. a) $t = 50 \triangle t$, b) $t = 100 \triangle t$, c) $t = 150 \triangle t$, d) $t = 260 \triangle t$.

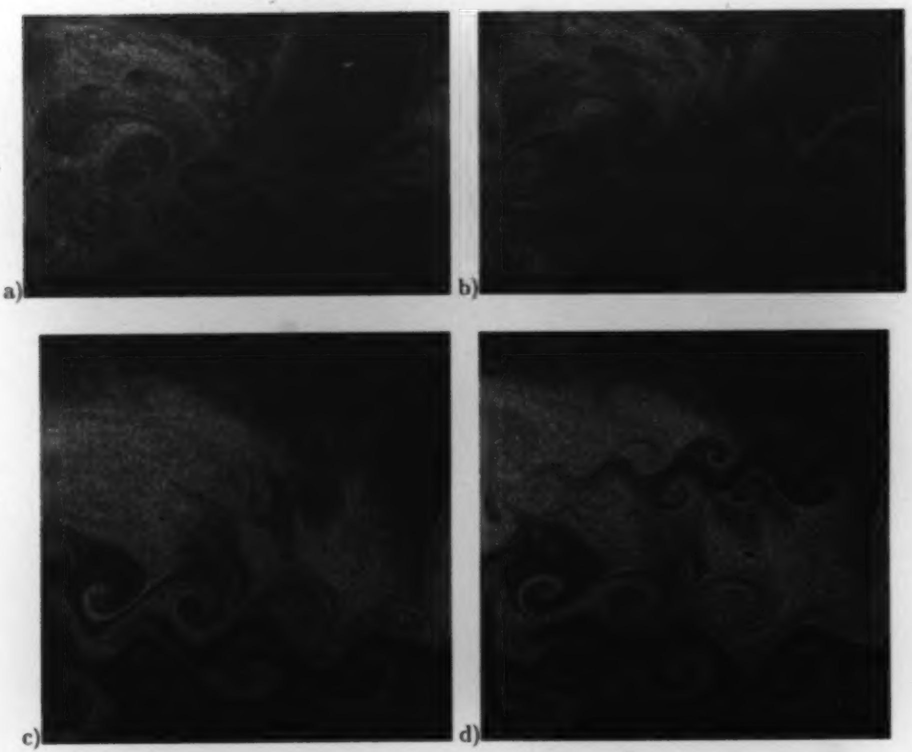


Figure 5.21: Karman wake of a flat plate, a,b) real experiments, c,d) numerical solution with the sinusoidal input parameters, c) $A = 0.1$, $f = 4$, $t = 45 \,\Delta t$, d) $A = 0.05$, $f = 6$, $t = 90 \,\Delta t$.

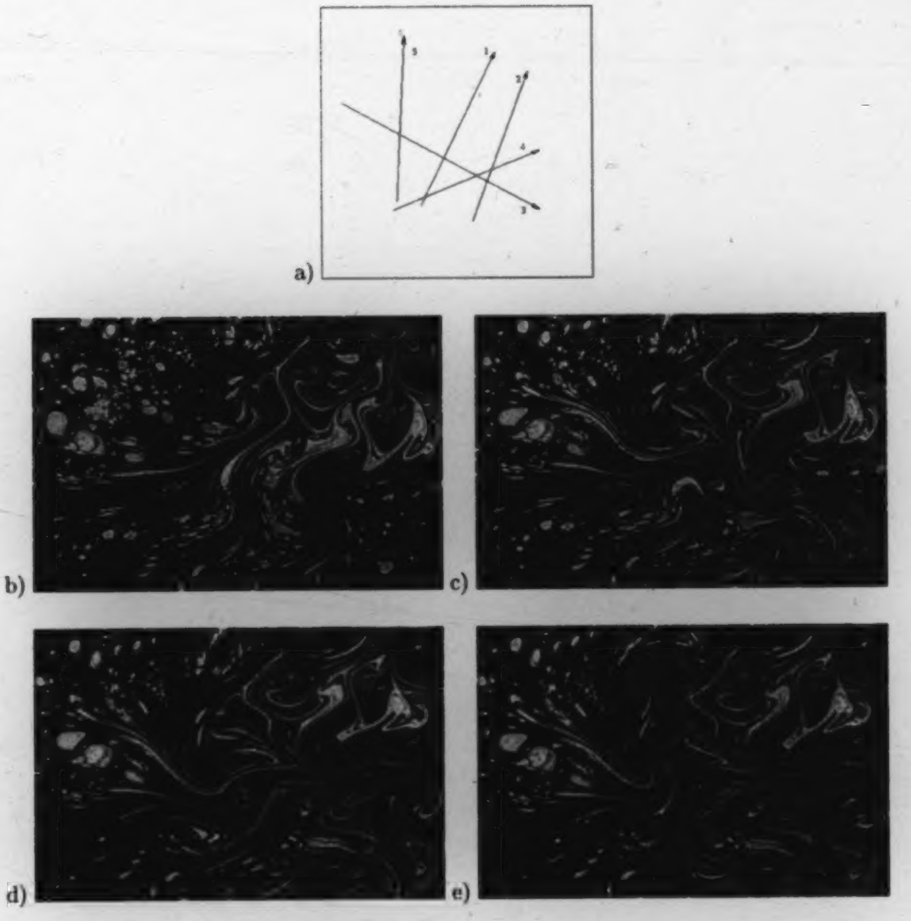


Figure 5.22: **Flat plate wakes in a higher viscosity fluid, real experiments. a) Stylus strokes, b-e) pattern formation after each stroke.**

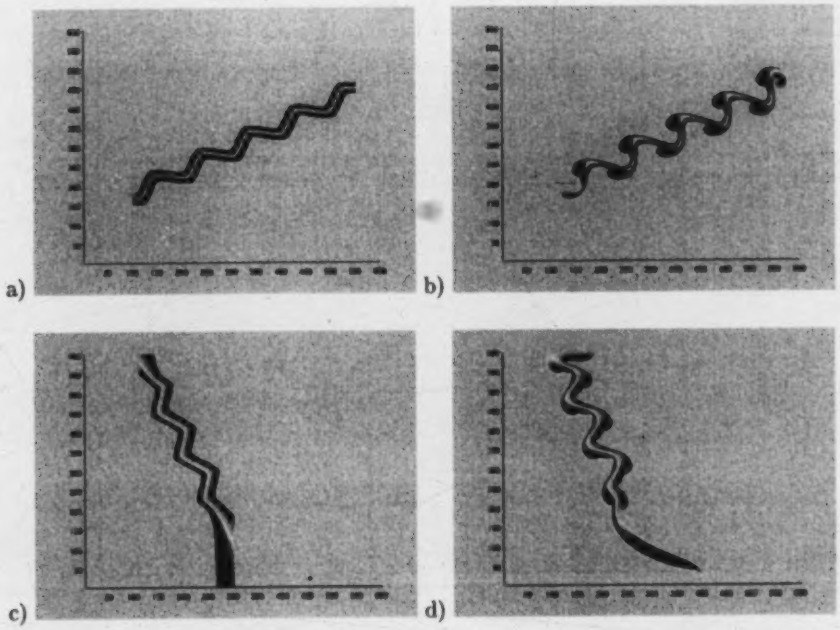


Figure 5.23: Flat plate wakes in a higher viscosity fluid, numerical solution: level set data evolution for two different strokes.

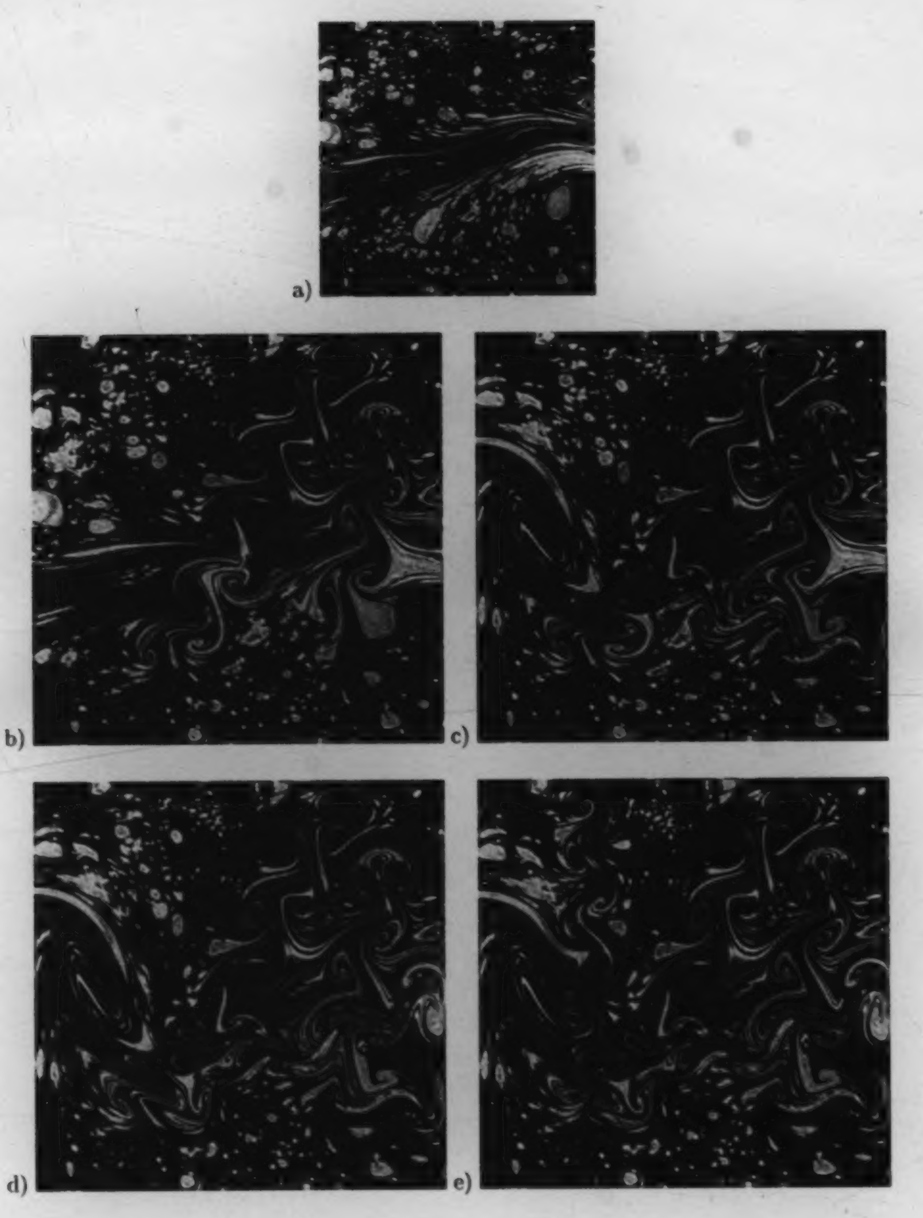


Figure 5.24: **Flat plate wakes in a higher viscosity fluid, numerical solution: a) initial configuration , b-e) advection patterns after five strokes.**

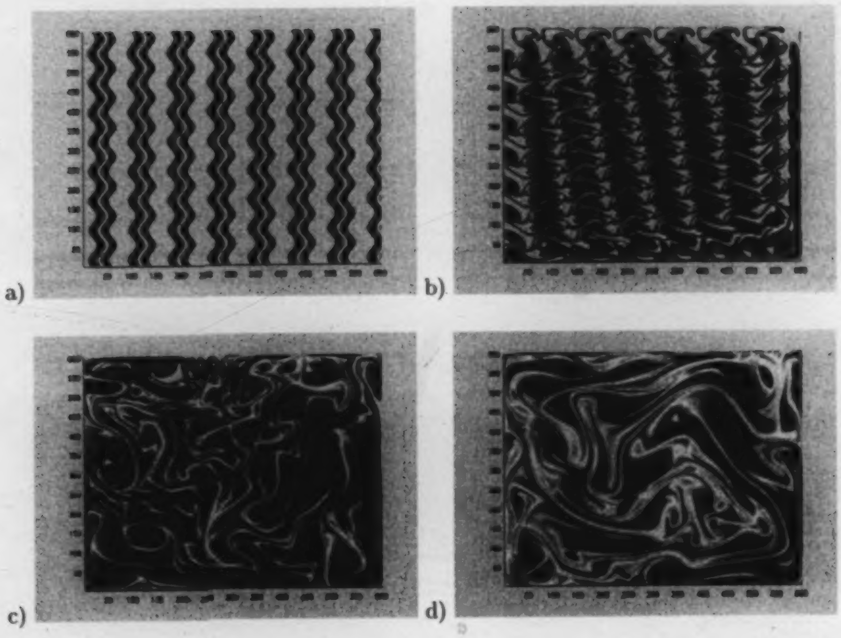


Figure 5.25: 2D grid turbulence: level set evolution. a) $t = 0$, b) $t = 200 \triangle t$, c) $t = 450 \triangle t$, d) $t = 700 \triangle t$.

4. 2D Turbulence: Topological Transitions

The level set method also handles the topological transitions automatically. I produced grid turbulence by applying multiple vortex sheets which are based on the sinusoidal initialization. Vortex sheets are kept a distance apart to allow the development of opposite signed vortices. In Figures 5.25, and 5.26 we observe the interaction of vortices and the evolution of turbulence; small eddies slowly merge into larger vortices. This conforms to the 2D turbulence theory [93]; the results can be compared to 2D turbulence experiments as well [106].

5. Point Vortices

One can also easily create point vortices by using a distance based level set data:

$$\varphi(\mathbf{x}) = min(\|\mathbf{x} - \mathbf{x}_i\|). \tag{5.25}$$

Figure 5.27a shows the smoothed level set data produced in this way, where opposite vorticity signs are assigned to each half of the plane, and Figure 5.27b shows the corresponding advection patterns.

6. Further Improvisation: Vortex Sheet Model as a Tool for Artistic Effects

By making use of the properties the level set transformation and the Poisson equation, this technique can also be used to create other interesting effects. For example periodic patterns can be designed

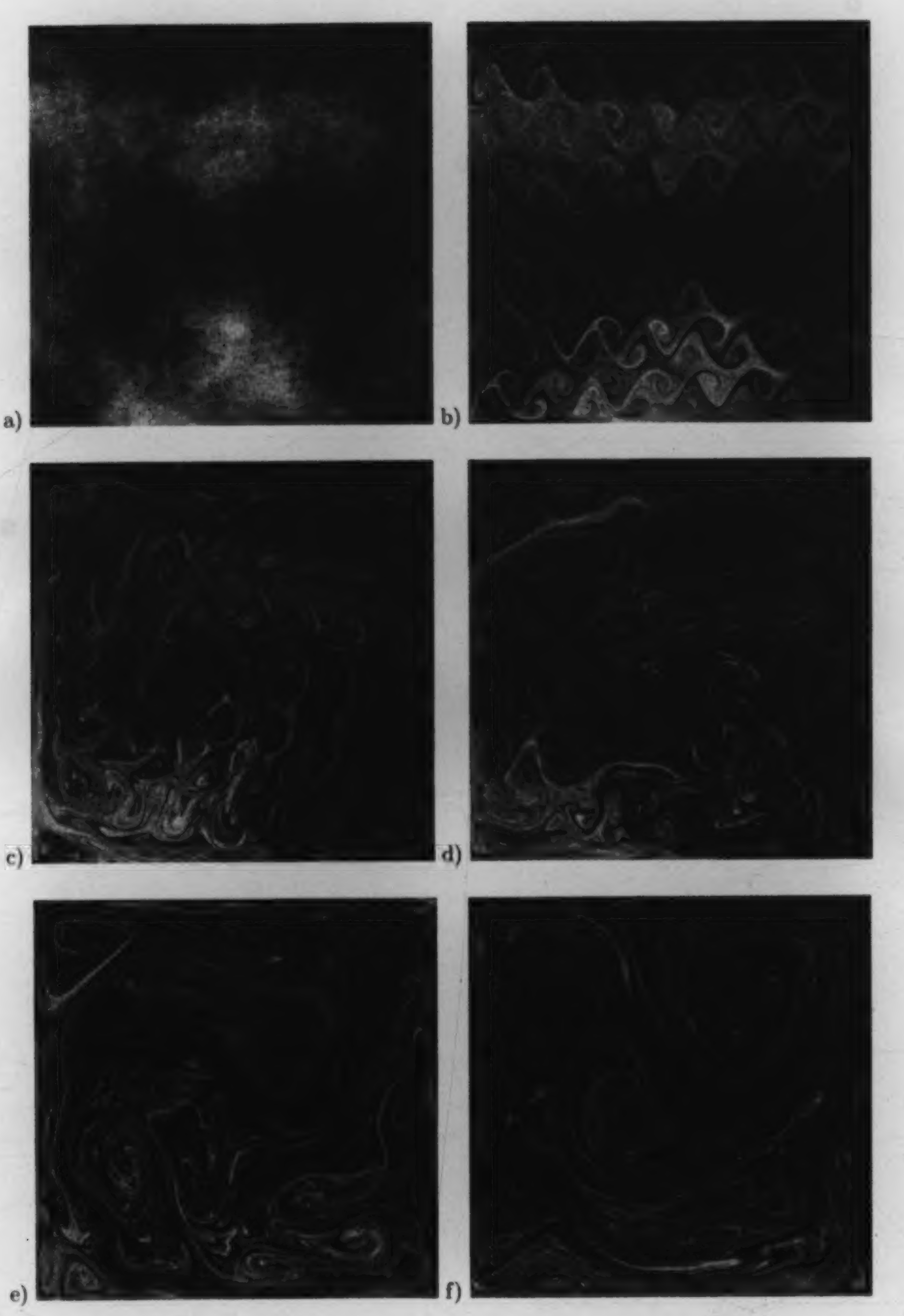


Figure 5.26: 2D grid turbulence: advection patterns: a) $t = 0$, b) $t = 100 \triangle t$, c) $t = 250 \triangle t$, d) $t = 350 \triangle t$, e) $t = 600 \triangle t$, f) $t = 845 \triangle t$.

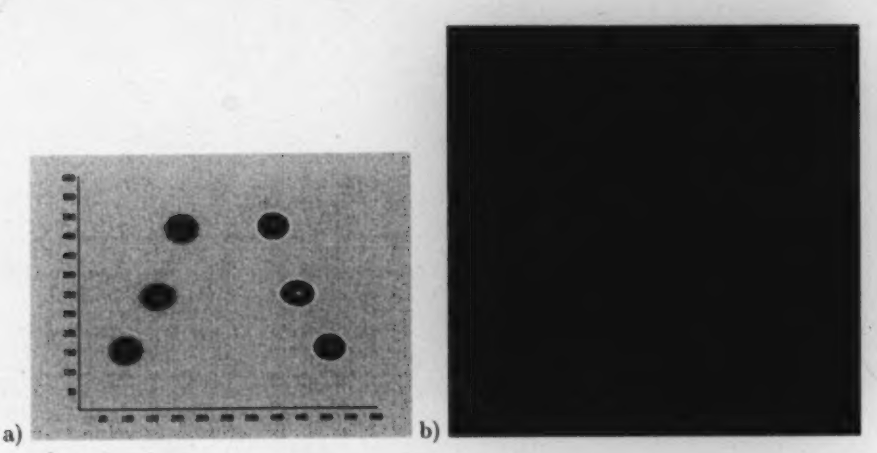


Figure 5.27: Point vortices: a) Level set data, b) Advection patterns.

easily using certain level set properties. Chaotic wave patterns shown in Figure 5.28 are obtained using periodic data. In this experiment, a typical sinusoidal level set input is defined as follows:

$$\varphi(\mathbf{x}) = y + 0.01\sin(2\pi x), \tag{5.26}$$

and the periodicity is obtained by using the transformation,

$$F(\varphi) = \frac{\pi \sin(\frac{\pi\varphi}{\delta_\epsilon})}{2\delta_\epsilon^2}, \tag{5.27}$$

where $\delta_\epsilon = 0.2$. The advection patterns are obtained from the repeated application of this comb in vertical and horizontal directions.

Furthermore, level set thickness (also considering its sign transition) can be adjusted to control the vortex sheet interaction in different ways. The patterns in Figure 5.30 are obtained from the vortex sheet interaction, the first part is obtained similarly in the previous example. Due to the interaction of negative and positive layers, the vortex sheet deforms first in horizontal direction, then a vertical transformation takes place. The level set data is shown in Figure 5.29.

One can also easily simulate the traditional comb patterns (used in *Ebru* designs) with this method. For example, in order to simulate the comb effect, I defined a periodic input based on a tangential function as follows:

$$\varphi(\mathbf{x}) = A\tan(f\pi x + a_0). \tag{5.28}$$

Here frequency, f can be used to determine the number of pins of the comb. The scaling factor, A, can be used to adjust the size of the pins, and the parameter, a_o which indicates a shift value, can be used to determine the location of the comb. Figure 5.31 shows the properties of the tangential function, sharp peaks in sign transition when applied to a sinusoidal smoothing function creates the comb effect (deformation along the pins). One can also add different perturbations to the tangential function to design different paths for the comb. For example let's define the following input to obtain

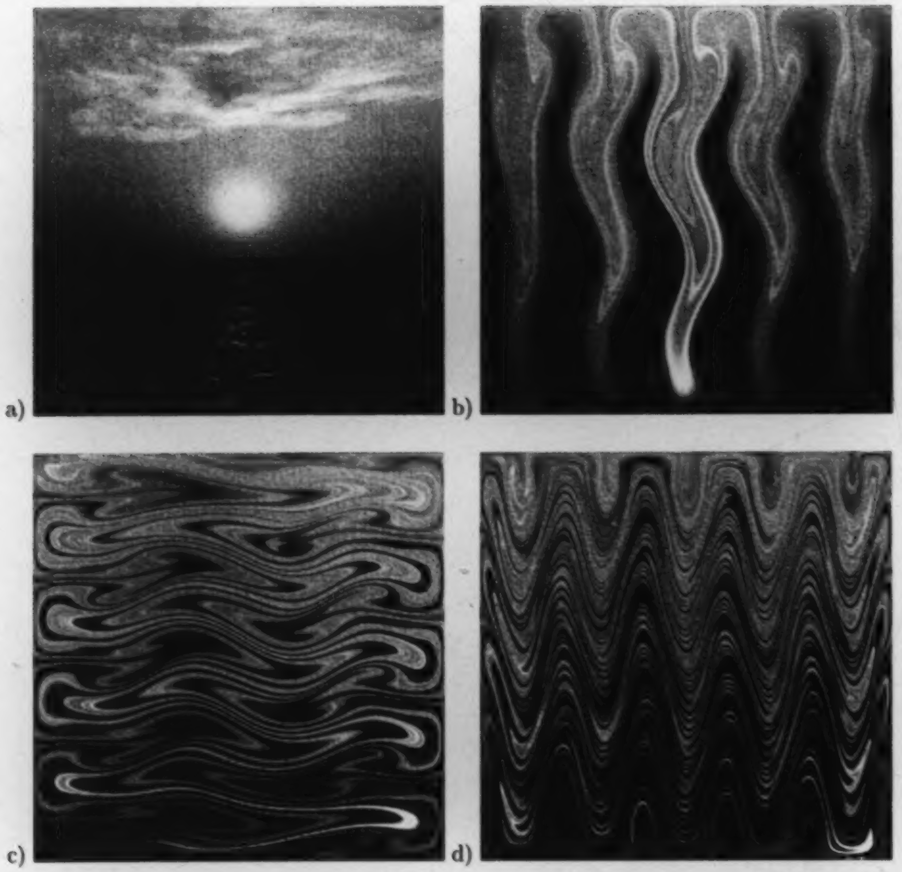


Figure 5.28: Periodic level set, horizontal/vertical wave patterns.

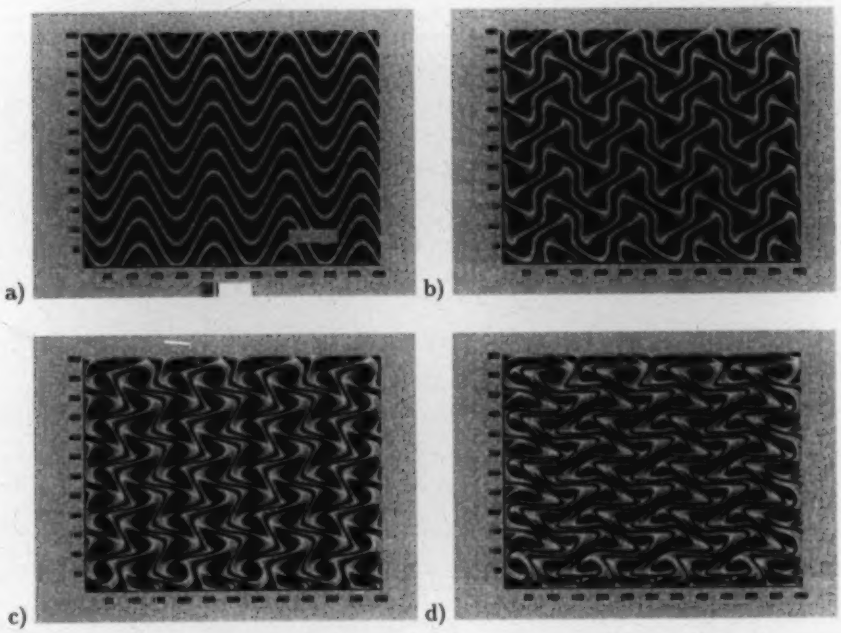


Figure 5.29: Vortex sheet interaction: level set data. a) $t = 0$, b) $t = 3 \Delta t$, c) $t = 7 \Delta t$, d) $t = 12 \Delta t$.

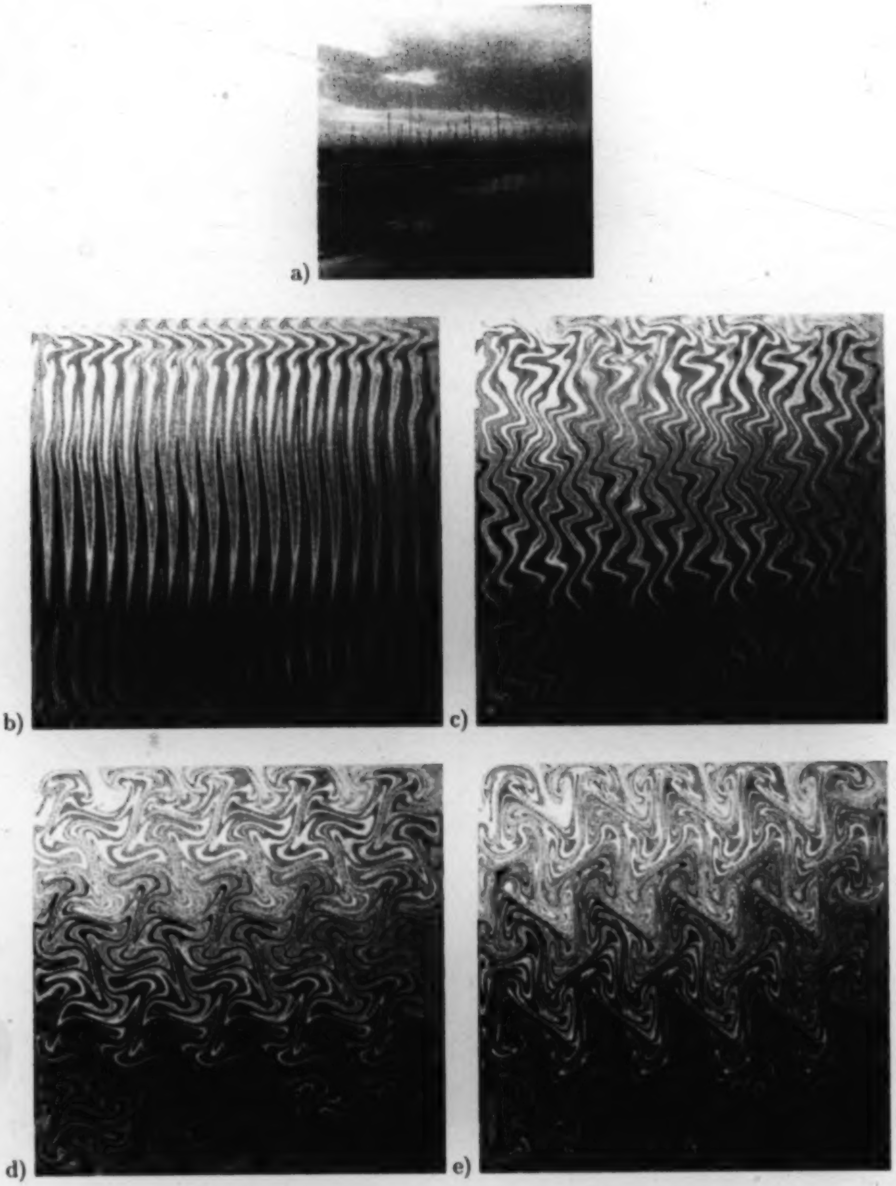


Figure 5.30: **Vortex sheet interaction: advection patterns.**

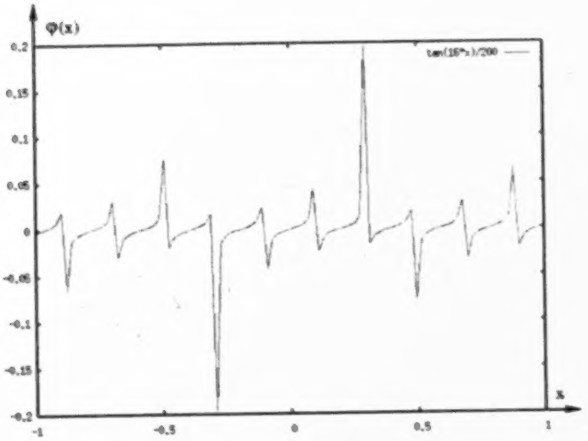


Figure 5.31: Level set (φ) initialization with a tangent function.

a sinusoidal comb path:

$$\varphi(\mathbf{x}) = A\tan(f\pi y + \sin(\pi x)). \tag{5.29}$$

Figure 5.32 shows the comb effects produced in this way. Note that, vortex sheet evolution is usually prevented in comb patterns. By using these combs, two of the classic marbling patterns are reproduced as shown in Figures 5.33 and 5.34, following the instructions given in the marbling book [109]. For *Gel-Git* patterns, instead of two-step interlaced rakes, sinusoidal input is used to obtain the same effects with one pass.

As mentioned before, Fractal Design Painter [54] provides a tool to create those patterns, however with predefined functions according to certain marbling recipes. With the level set technique the artist can design any type of comb pattern either freehand or by making use of mathematical functions. One can think of several other ways to design the level set data, for example real images with appropriate interface properties can be used for this purpose.

5.2.3 Multiscale Simulation of Turbulence and Fluctuating Motion Based on a Mesoscale Model

My goal was to simulate the complex, dynamic and unpredictable flow behaviours observed in some marbling styles especially in Suminagashi techniques. In order to simulate these types of flows which can not be simulated by viscous Navier–Stokes models, I developed a multiscale stochastic fluid model based on mesoscale dynamics. This model is based on the assumption that mesoscopic fluctuations are carried out to larger scales. Their effect at different levels, which could be in the form of mild oscillations or turbulent eddies are modeled by using an averaging method. First let's present some background information on the statistical properties of turbulence and briefly review

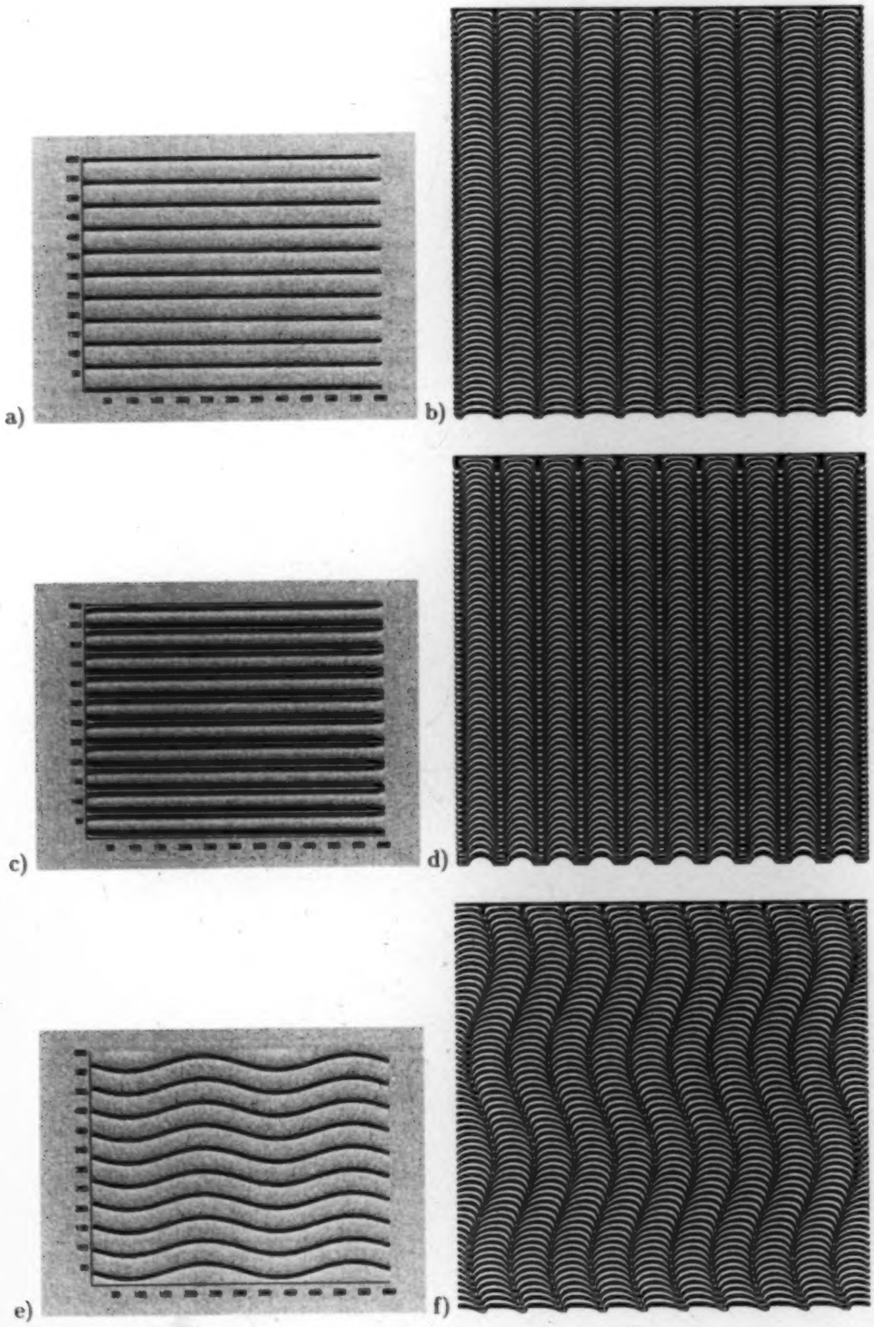


Figure 5.32: Comb effect with periodic level set based on a tangent function: a,b) $A=0.005$, c,d) $A=0.01$, e,f) sinusoidal path

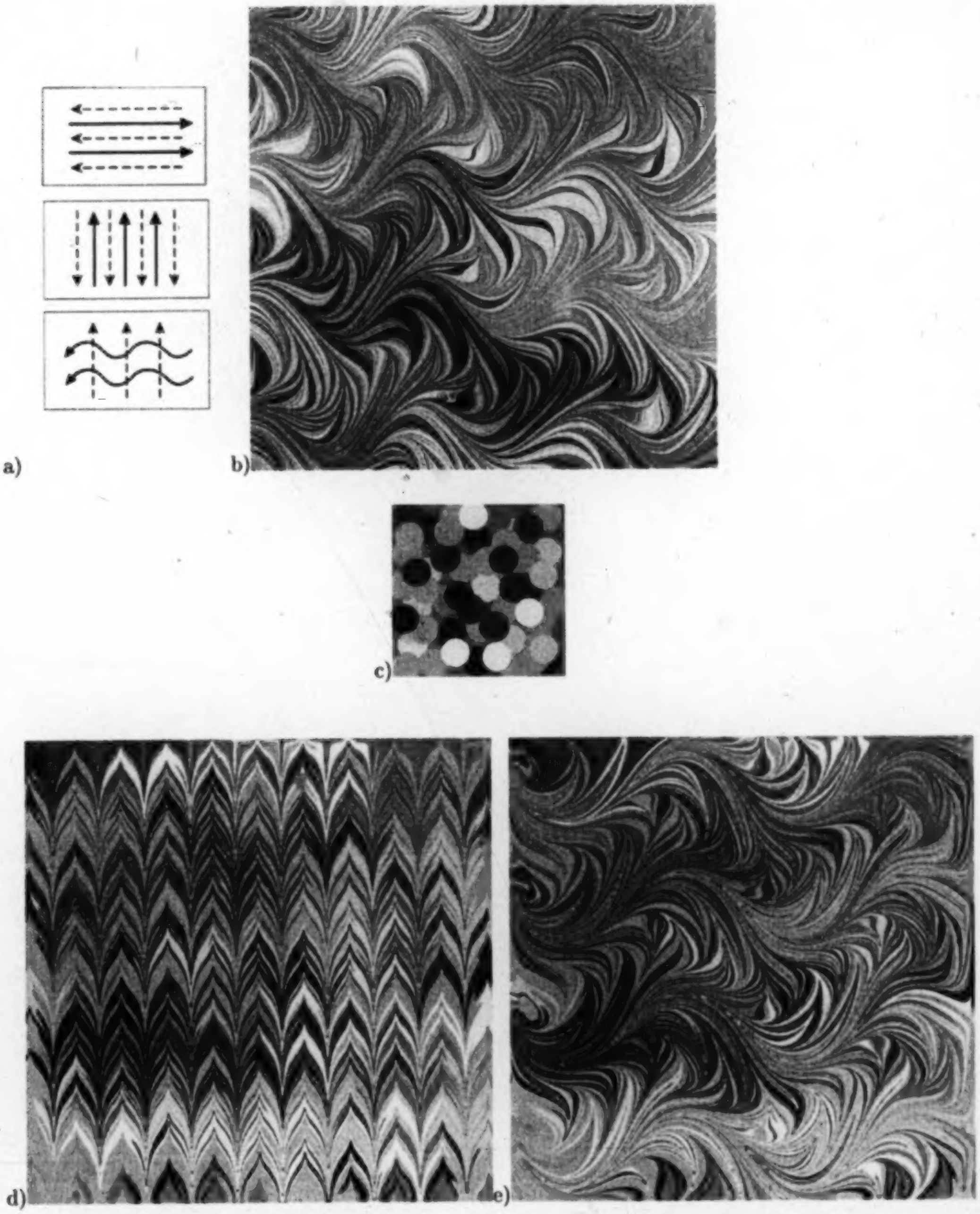


Figure 5.33: Chevron pattern. a) Comb instructions [109], b) Real design [109], c) Initial configuration guess for the numerical solution, d,e) Numerical results: two frames from the animation

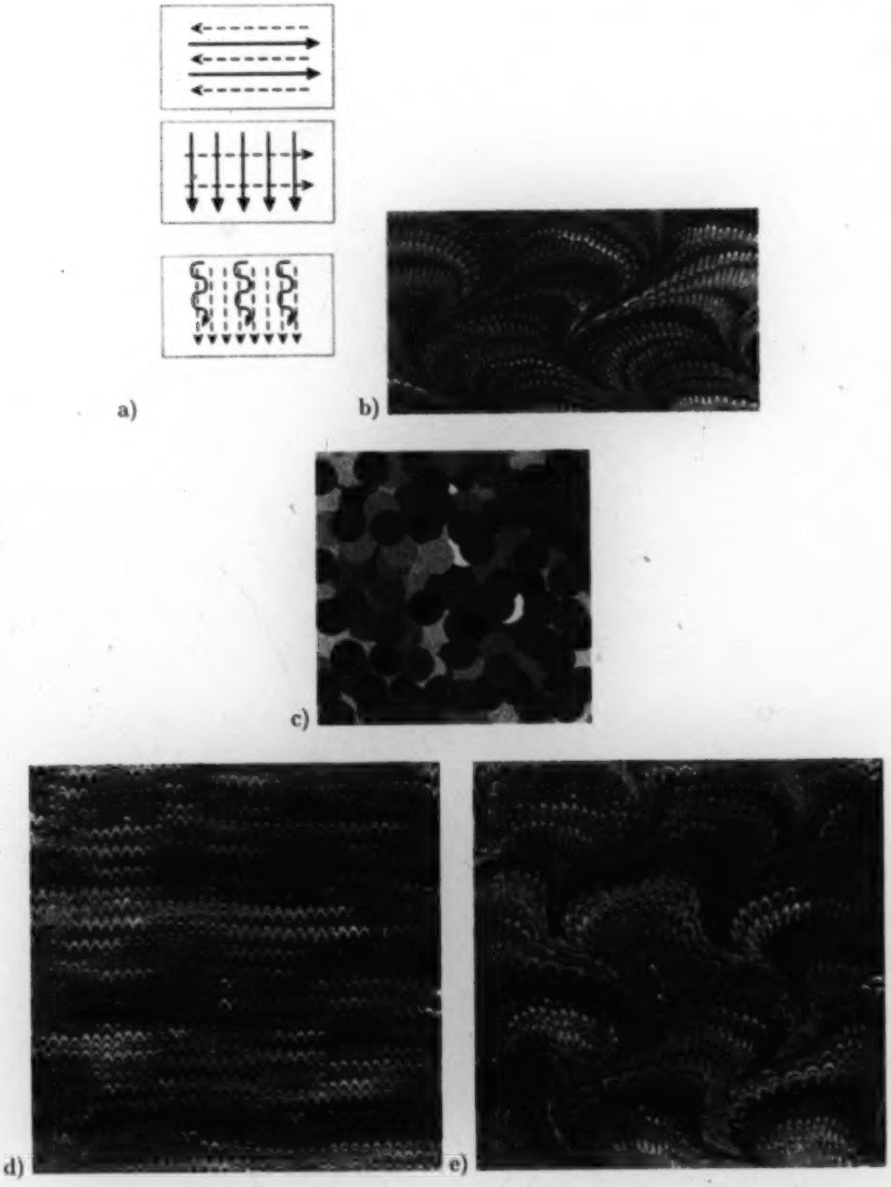


Figure 5.34: Bouquet pattern a) Comb instructions [109], b) Real design [84], c) Initial configuration guess for the numerical solution, d,e) Numerical results: two frames from the animation.

the conventional turbulence models.

5.2.3.1 Statistical Properties of Turbulence

In theory, from a generally accepted point of view, Navier–Stokes should be sufficient by itself to model all fluid behaviours. However as Reynolds number increases it becomes very difficult to model the details at small scales mainly due to numerical limitations. (In vortex sheet simulation, we have seen some of the difficulties in numerical modeling when the flow is sensitive to small fluctuations). Even if the exact solution of Navier–Stokes at all scales can be obtained, whether or not a deterministic system can fully describe the chaotic nature of turbulence is still debatable. For these reasons, turbulence studies, have mostly been based on statistical approaches and direct numerical simulation (DNS) of turbulence by Navier Stokes have been limited to a certain range of Reynolds numbers.

Obtaining the relations described by the two basic laws of turbulence from the fluid equations has been one of the biggest challenges in turbulence modeling. Kolmogorov's well-known 1941 study plays an important role in turbulence theory as it was the first to derive these relations from Navier–Stokes equations. Kolmogorov's theory [59, 57] describes the velocity correlation in homogeneous isotropic turbulence at infinite Reynolds number. Velocity correlation depends on the two-third power of the distance, r, and energy dissipation is given by $(\epsilon r)^{2/3}$ where ϵ indicates the energy dissipation rate. According to Kolmogorov's universality assumption, statistical properties at small scales depend on the scale, r, and energy dissipation rate (ϵ). Kolmogorov [57] derived these results from Navier–Stokes by using the Karman-Howarth relation which describes the evolution of velocity correlation as follows:

$$\frac{\partial B_2}{\partial t} = \frac{2\nu}{r^4}\frac{\partial}{\partial r}\left(r^4\frac{\partial B_2}{\partial r}\right) + \frac{4\epsilon}{3} - \frac{1}{3r^4}\frac{\partial}{\partial r}\left(r^4 B_3\right), \tag{5.30}$$

where B_2 and B_3 are the second and third order structure functions. This equation, which also describes the energy dissipation is obtained by averaging the randomly forced Navier–Stokes equation, in other words the Navier–Stokes equation represented by variables which include fluctuating components such as $v + v'$, under the local isotropy assumption. The energy spectrum of Kolmogorov which follows from the two-thirds law is given by:

$$E(k) \sim (\epsilon r)^{2/3} k^{-5/3}, \tag{5.31}$$

where k represents the wave number. In 2D, although this relation still applies to smaller scales, above a certain wave number enstrophy cascade (inverse cascade) becomes effective. Thus energy cascades to large scales and enstrophy cascades to small scales. Although this is still not a completely resolved discussion and there are various theories, the energy spectrum in 2D turbulence is commonly described as:

$$E(k) \sim (\epsilon r)^{2/3} k^{-5/3} \quad k < k_{\text{inj}}, \tag{5.32}$$

$$E(k) \sim (\epsilon r)^{2/3} k^{-3} \quad k_{\text{inj}} < k < k_d, \tag{5.33}$$

where k_{inj} indicates the energy injection scale (which can be considered the grid scale in 2D) and k_d denotes the dissipation range. The second part (Eqn. 5.33) describing the inverse cascade which is specific to 2D turbulence was derived by Kraichnan [98] and Batchelor [4] based on enstrophy conservation.

These are some of the most important "exact" results defining the properties of turbulence under homogeneous and isotropic conditions. (***Homogeneous*** implies that the statistical properties do not change with position and ***isotropic*** means rotation invariant.) Note that, homogeneous isotropic turbulence is an ideal situation which rarely happens in nature. These laws are used as a theoretical condition to prove the acceptability of a turbulence model. In other words, a turbulence model is expected to satisfy these laws in order to be realistic.

5.2.3.2 Conventional Turbulence Modeling

The most common turbulence methods [16], based on the statistical averaging idea are Reynolds Averaged Navier Stokes (RANS) calculations and Large Eddy Simulation (LES). RANS is based on the assumption that flow variables in the Navier–Stokes equations have a mean and a fluctuating component $(x + x')$. By time averaging (or ensemble averaging) the Navier Stokes equations with these variables, a set of equations are derived which describe the motion of the average flow together with the effect of fluctuations on the mean flow. This additional stress is called the Reynolds stress tensor (R_{ij}); basically it represents the correlation between velocity fluctuations:

$$R_{ij} = -\rho\overline{u_i'u_j'} = \mu_e\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{5.34}$$

where the subscripts i and j indicate different spatial coordinates, overbar indicates averaging and μ_e denotes the eddy viscosity which describe the turbulence characteristics.

The main purpose of the averaging models (both RANS and LES) is to formulate the extra terms arising from the averaging in terms of the mean flow values, that is to obtain closure models. This is still an open problem, various approaches have been proposed, mostly based on empirical assumptions according to the characteristics of the flow being modeled. It is very difficult to develop one model which provides a general solution to all the possible complexities in turbulent flows. Basically in Reynolds averaging closure models eddy viscosity is related to the mean velocity gradients. Thus RANS calculation is considered more appropriate if mean value is of concern; RANS closure models (mostly being limited to specific flows) have not been able to model complicated separated flows. LES is a more advanced approach which is developed to model the higher levels of complexity that are important for turbulence. In LES, spatial averaging is applied to the Navier–Stokes equations in order to filter out the scales below the filter width and resolve larger scales. The averaged cross terms $(\overline{u_i u_j})$ coming from averaging the nonlinear advection term need to be determined. At this point, the flow variables are decomposed into a mean and a fluctuating component and the averaged nonlinear term is represented using these variables. As a result, an extra term, the subgridscale stress, is obtained. The subgridscale term is used to model the effect of small scale fluctuations on larger scales. The extra term is more complicated than the one which is obtained from Reynolds averaging; since local isotropy assumption is not used as in RANS, most terms are not canceled out. The purpose of the closure models is to determine the subgridscale terms based on the energy cascading properties of turbulence. The main difference of LES from RANS is that the filter width can be controlled to resolve various scales. In this sense LES is considered to provide higher accuracy as it allows the approximation of Direct Numerical Simulation (DNS) as the grid resolution is increased. However LES has a higher computational cost. In fact, which approach is more appropriate for which applications is still under discussion.

5.2.3.3 Turbulence Modeling in Computer Graphics

Graphics work in turbulence simulation usually follow a practical approach focusing more on the visual aspects rather than physical accuracy. Most of these studies aim at modeling gaseous phenomena such as smoke, clouds, gas etc. Stam et al. [138] used an external random force in the momentum equation, which is generated by Fourier synthesis, to model small scale wind fields. Despite its convenience in adding small scale details in gaseous flows, this is rather a simplistic approach for turbulence modeling as the random force doesn't have a physical meaning. Lately in [125], Rasmussen et.al. used this type of random forcing (which is based on Kolmogorov energy spectrum) to build the third dimension of a periodic velocity field. More specifically, two-dimensional incompressible fluid solution was combined with this random field to obtain highly detailed three-dimensional velocity fields (which include small scale eddies) to simulate smoke and explosions.

In [52] Foster and Metaxas used the convective heat transfer model to simulate the temperature driven turbulence in gasses. In this model, an additional term representing the thermal buoyancy is included in the momentum equation. The evolution of temperature is described by the advection-diffusion equation. Fedkiw et.al. introduced the "vorticity confinement" method in [43] which was used to model smoke. Vorticity confinement method [143] is used to reduce the effect of diffusion error in vortex development in inviscid (Euler) equations. An additional source term is included in the momentum equation to compensate for the loss caused by numerical diffusion. This term, which depends on the local vorticity, is designed to confine the vorticity to a local region by continuously feeding the vorticity information back into the equation as it gets diffused. Vorticity confinement has a local effect, for more realistic and global behaviours a general turbulence approach is required. Note that, vorticity confinement has been used in other models in graphics [120, 44] to increase the turbulent behaviour in fluid equations.

These works were actually based on modeling fire and flames. One of the earliest work in this area was Stam&Fiume's model [139]; they defined a simple temperature based source term in the advection diffusion equation to model flames. Fluid vectors were obtained from their "wind field" model. In a recent model for smoke simulation [42], Fattal et.al. defined an external force in the fluid equations based on density gradient to add turbulence in a controlled way; this is used to direct the evolution of smoke in the direction of concentration gradient.

5.2.3.4 Multiscale Fluid Model for Marbling

In conventional turbulence models fluctuations do not appear in the solution, their effect on the mean flow are formulated through separate equations. I propose using mesoscale fluctuations as the driving force of turbulence: in this method, fluid equations which include the fluctuating forces are filtered recursively with a spatial approximation method (kriging) in order to simulate the effect of fluctuations at different levels. Thus fluctuations enter the solution directly. It should be mentioned that, being derived intuitively based on the general concept of statistical averaging, this method has not been tested for its validity with respect to the laws of turbulence in theory. I will emphasize its relevance to some of the unconventional turbulence models as well as the traditional ones, and its properties which contradict some of the physical rules. However a theoretical study in this respect constitutes an interesting research topic for the future. In practice, on the other hand this method can be used to model a variety of flows, some of which are very close to real examples. This model is developed to provide the user a flexible environment where she has full control of the fluctuations to create a variety of flows and different effects. Although these recent mesoscale models constitute a potential for turbulence modeling, they haven't been used for turbulence modeling to my knowledge.

The method basically involves the following steps.

1. Solve the momentum equation with the mesoscale fluctuations.
2. Solve Poisson equation to enforce incompressibility
3. Filter the velocity vectors obtained from step 2 using kriging approximation.

Each step can be repeated recursively. The number of repetitions for each step, together with the fluid parameters determine the scale of averaging and the characteristics of the flow.

First let us recall the addition of fluctuations in the fluid equation (from Chapter 4); they are not just arbitrary external forces added to the momentum equation, but they are related to the underlying microscopic description of the fluid. In other words fluid elements represent thermodynamically consistent quantities and they are designed to conserve momentum. DPD based methods have been used for modeling complex fluids in their original Lagrangian formulation [7, 121, 41]. In these methods, mesoscale fluctuations are assumed to be averaged out at the macroscopic level. And their relation to the macroscopic level is defined through the fluctuation-dissipation theorem which is based on balancing the random and viscous forces. It is a characteristic of the Brownian motion, that it appears in a short range. However not in all cases the random, fluctuating motion tends to stay at small scales. For example, certain colloidal suspensions, particularly which are subject to surface ionization, as in marbling, are likely to produce such long time effects. However these discussions cannot be resolved within the scope of this thesis. In this model, it is simply assumed that the effect of mesoscale fluctuations are carried to larger scales, creating fluctuating, complex flow behaviours, if not turbulent. To achieve this, basically the step size of the random walk is exaggerated while still conforming to the fluctuation-dissipation theorem. I introduced a magnifying parameter, α, in the magnitudes of the fluctuations:

$$\sigma^2_{ij\|} = 2\sigma^2_{ij\perp} == \frac{4\gamma k_B \overline{T_{ij}}\alpha}{r_{ji}}. \tag{5.35}$$

Recall that, γ represents the viscosity (friction coefficient) and $\overline{T_{ij}}$ denotes the average temperature. This modification can also be thought of as increasing the temperature. The parameter, α, is defined by the user; it will also be used to apply conditioning.

In this model, characteristics of the original mesoscopic fluctuations have been altered mainly in two aspects:

- By changing the magnitude of the random fluctuations, so that they do not represent conventional Brownian motion anymore but they can affect larger scales.

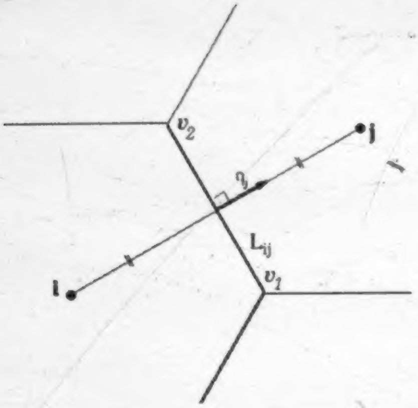


Figure 5.35: Voronoi cell structure.

- By making the random forces conditional their isotropic and homogeneous properties are changed.

In Chapter 3, the suitability of the Voronoi mesoscale models [41, 46] for scaling was discussed. Given this, I used kriging to average the velocity fields which include random fluctuations. This filtering process assumes that the correlation of the velocity field is defined by the kriging variogram. Characteristics of the correlation function such as the cut off range, and decay rate determine the velocity variation similarly in Kolmogorov's theory for homogeneous isotropic turbulence. It should be mentioned that, first kriging studies were actually intended to obtain the average hydrodynamic behaviours from the velocity variogram, in geological systems. However later, kriging turned out to be just an approximation method which incorporates the statistical properties of the data. I should also note that, I haven't encountered a similar usage of kriging as a filter for turbulence.

Random forces are generated according to a Gaussian distribution, uncorrelated in time. In order to enforce the symmetry conditions, let's define the following algorithm to apply the random forces:

```
For each neighbour (j) of cell (i) (Figure 5.35)
  if $\mathbf{R}_{ji}$ is defined
    $\mathbf{R}_{ij} = \mathbf{R}_{ji}$
  else
    Assign a new random force to $\mathbf{R}_{ij}$
```

This implementation shows that the random forces are actually defined on the cell intersections rather than the cell centers. Also the forces are generated on a first-come first-served basis, that is they are assigned in the scanning order. This was my interpretation of the mesoscale fluctuating motion in the Eulerian frame.

Conditioning can involve, for instance, dependency on velocity or density variations or external boundaries. Realistic turbulence is triggered by forces produced by fluid boundary interactions or

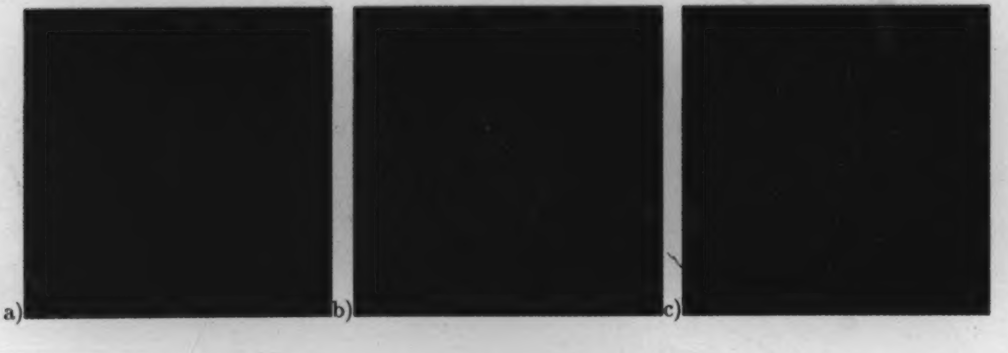


Figure 5.36: Averaging of vectors. a-c) Random fluid steps.

by interface instabilities, etc. In my method, mesoscale fluctuations are used to inject the driving energy for turbulent or fluctuating motion. They could be representing external grid forces or fluid based fluctuations.

Note that, in case of conditional forcing , properties of the mesoscale fluid such as the fluctuation-dissipation relation or the symmetry conditions are preserved, only the magnitude of the random force is made conditional.

When fluctuations depend on velocity variation, a random force is generated between two neighbours only if there is a relative velocity between them. This is implemented as follows:

```
$\theta = 0$
if $(\alpha > 0)$ $AND$ $(|\mathbf{u}_{ij}| > \theta)$
  Apply the random force
```

with $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$, and θ represents the maximum limit for the relative velocity below which no random force is generated. Figure 5.36 shows the evolution of the vectors around a velocity perturbation in the center. Note that, random forces (their magnitude and spreading level) are exaggerated here; each fluid step is run with the parameter values, $\gamma = 0.1$, $\alpha = 0.1$, $\theta = 0.001$, *initial_velocity* = 2.0, *divergence_steps* = 1. Figures 5.37d-i show the smoothing levels from repeated averaging of the vectors from the third step.

5.2.3.4.1 User Control

Now let's review the controlling parameters of the system. First of all, it should be pointed out that the smoothing process might affect the following fluid properties:

1. The incompressibility of the velocity vectors since the interpolation with smoothing does not guarantee conservation.
2. Momentum conservation.
3. The balance between the fluctuating and viscous forces imposed by the fluctuation dissipation theorem.

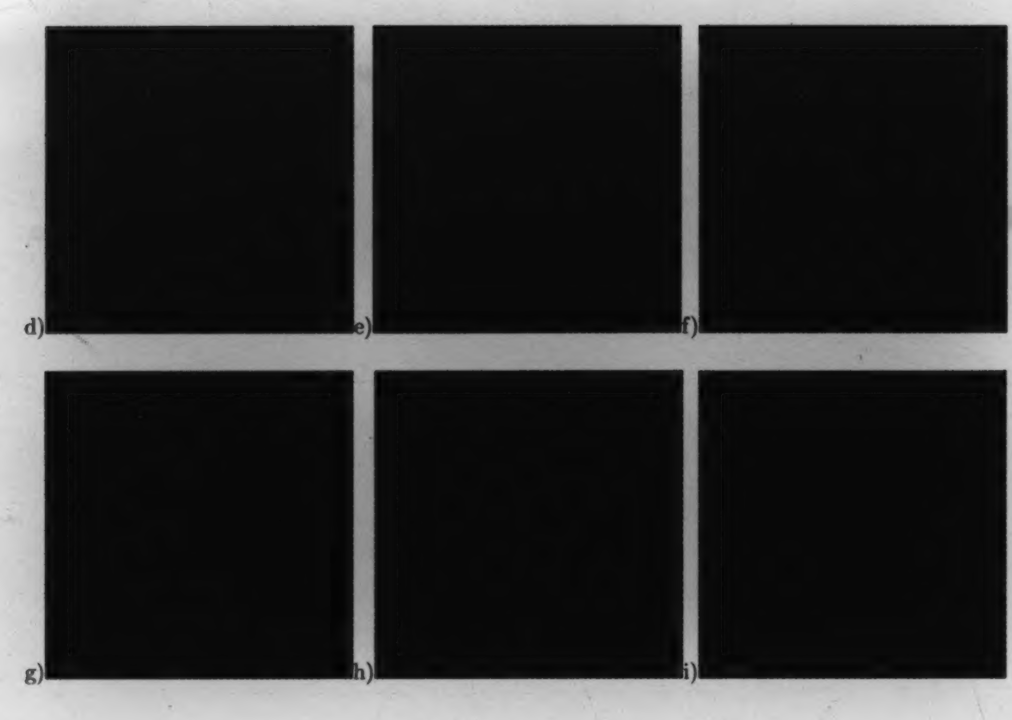


Figure 5.37: Averaging of vectors: d-i) Smoothing steps.

The last two do not constitute a critical issue regarding the physical accuracy of this application. As for the incompressibility, this effect will be used to control the divergence properties of the fluid, considering that, in real marbling, most flows are not purely rotational.

The first two parameters defining the fluctuating flows are viscosity and the properties of the random forces. According to the fluctuation-dissipation relation, the effect of the random forces is connected to the effect of the viscous forces as follows. Increasing the viscosity causes the random fluctuations to become more effective. On the other hand, at lower viscosities stronger random forces must be applied to create a more powerful influence on larger scales in which case the flow usually becomes more turbulent.

Given these, the number of times each step is repeated in the algorithm has the following effects on the flow:

1. Considering that the fluctuations take place at smaller scales, repeating the momentum equation more than once can be used to predict the long time behaviour of the random forces. In more practical terms this step controls the spreading of the randomly forced velocity vectors depending on the condition imposed.

2. Divergence field, that is, the accuracy level of the incompressibility, will be controlled by adjusting the maximum steps for the convergence condition in the incompressibility solution. Recall from Chapter 4 that increasing the maximum convergence limit provides a closer approximation to the continuity constraint. Divergence field adjustment will also be used to compensate for the loss of information caused by repeated smoothing.

3. Number of smoothing steps determines the scale of the effect of fluctuations, that is, the strength of their influence on larger scales. Also depending on the magnitude of the random forces, adjustment of the smoothing level will create a range of effects from smooth oscillations to turbulence.

The parameters corresponding to the repetition of these steps will be denoted by *momentum_steps*, *max_divergence_steps*, and *smoothing_steps* respectively.

5.2.3.4.2 Some Implementation Issues and Discussion

Currently only one nugget value is used in kriging smoothing; in order to increase the averaging level, same smoothing weights are applied recursively. Both the interpolatory and the smoothing kriging weights on the image grid are precalculated and installed in memory at the beginning of the simulation. This is done to save computation time. In case the external boundary point alters the Voronoi connections, that is, when high resolution boundary effect is required, only the weights in the local area, whose Voronoi connections are affected by the new point are updated. However I didn't find this necessary in the current experiments. Another issue is the temporal correlation of the random forces. As mentioned before, random forces are generated independently at each step. The correlation between the fluid steps depends on the random force/smoothing level proportion; if the fluctuations are not too strong, or the averaging is high enough, correlation is maintained. Note that, correlated fluctuations can easily be implemented in this system, for example they can be designed to decay exponentially in time according to a Gaussian distribution. Furthermore, properties of the kriging variogram, such as the range or the nugget level can be time dependent. These can be used to study physical aspects of the turbulence model.

At this point, I would like to emphasize that the proposed stochastic method is in a way similar to the Renormalization Group approach [169, 49] in which the large scale (that is, long time and large distance) behaviours of randomly forced Navier–Stokes equations are obtained by iterative filtering. Recursive filtering is done systematically though, to repeatedly eliminate the highest wavenumber modes corresponding to small eddies and relating their effect on the remaining modes. This is a computationally expensive process. RNG methods have been considered unconventional turbulence models, however lately more realistic versions have been developed in which the same RNG idea is used to obtain closure models ($RNG - k - \epsilon$) as in LES approach. In fact, my model can also be compared to the LES approach in the general sense that it aims at obtaining the effect of small scale fluctuations on larger scales through spatial filtering.

In conclusion, this system can be used to simulate a wide range of dynamic flows, including the complex fluid behaviours in most Suminagashi techniques as well as highly turbulent flows. Evolution of the flows will be unpredictable as in real life with no two flows being the same even for similar initial conditions. What is also appealing to a marbling application is that this model provides a flexible user environment where the user can control the effect of fluctuations at different scales. In the following section, I will demonstrate several applications of this method.

In the future, answers to some of the following questions regarding the accuracy of this system can be further investigated. For example, what are the outcomes of the modifications made to the original mesoscale model ? Concerning turbulence modeling, how well does this approach address the physical aspects of turbulence ? What's the connection of its statistical properties to the Karman-Howarth relation, or to Langevin based stochastic turbulence modeling ?

5.2.3.4.3 User Interface The user interface allows the control of basic flow parameters. These are boundary velocity, time interval, viscosity, magnitude of random fluctuations, number of momentum and smoothing steps. In addition incompressibility condition is optional. One can see in Figure 5.38 an example of the current user interface. Fluid equations and the transport model can be executed separately and step by step. In this way, the user can change the flow parameters during the simulation.

Boundary configuration is designed in the code. For certain boundaries, such as multiple point rotating boundaries the user can locate the boundary center on the screen. Or in case of point boundaries he can draw the boundary path on the screen, however due to speed limitations predefined paths are used in the current experiments.

The user can switch between three types of displays: flow vectors on the Voronoi grid, advected image on the fine grid, and interpolated velocity vectors on the fine grid.

Other than these, all the properties including the advection and interface model are designed in the code. Thus development of a high level user interface is essential in the future.

5.2.3.4.4 Experimental Results (Note that in all the experiments time interval is taken as $\Delta t = 0.7$; this was the stability limit for the Runge-Kutta time integration of the momentum equation.)

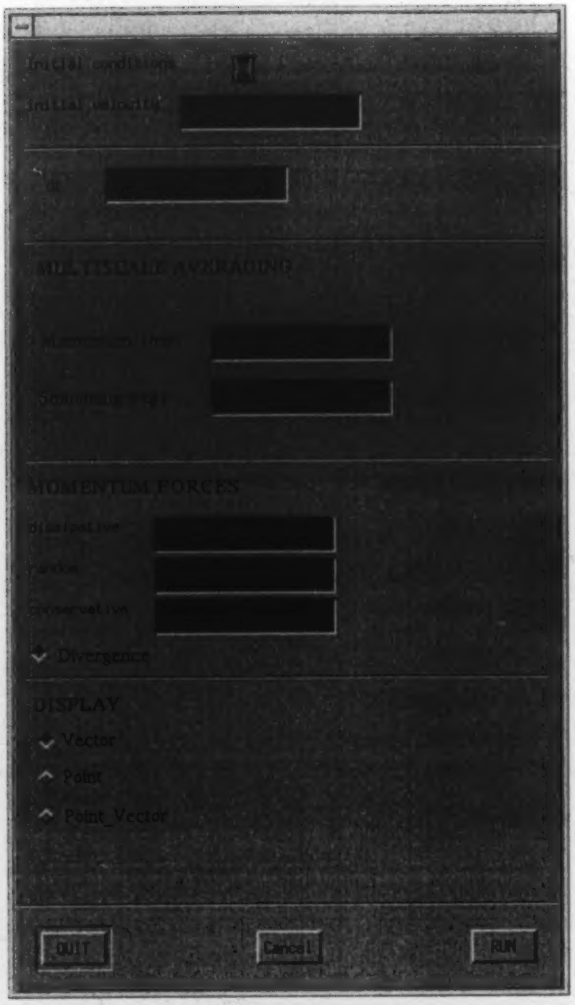

Initial conditions
Initial velocity
dt
MULTISCALE AVERAGING
MOMENTUM FORCES
dissipative
random
conservative
Divergence
DISPLAY
Vector
Point
Point_Vector
QUIT
Cancel
RUN

Figure 5.38: User interface for the multiscale fluid model.

1. Turbulence Around a Moving Boundary: Large Scale

The first two experiments demonstrate the turbulence caused by the boundary motion at two different scales. Both are based on the same boundary motion; a point boundary moves in the same predefined path but with different speeds. In the first experiment, the effect of fluctuations are strong, turbulent motion around the boundary spreads fast, on a larger scale. As the boundary moves it applies a velocity in the tangential direction to its path. The random force conditioning is based on the relative velocity between two neighbouring cells as explained above. This causes the flow evolving around the boundary to be more turbulent.

In order to make the fluctuating flow spread fast the limit of the velocity based conditioning is kept small, $\theta = 0.01$. Also the randomly forced momentum steps are increased, $momentum\_steps = 2$. Increasing the smoothing steps, ($smoothing\_steps = 7$) also causes the vectors to spread more and smoothly but more importantly it carries the effect of fluctuations to larger scales. Furthermore, to increase the effect of fluctuations, random force magnitude is increased according to the relatively high dissipation rate, $\alpha = 0.1$, $\gamma = 0.11$. Finally, in order to obtain a more vortical flow, divergence level is given by, $max\_divergence\_steps = 5$. Figure 5.39 shows the fast evolution of the large scale turbulent vectors as the boundary moves. In a short period of time the flow spreads to the whole grid. Figure 5.40a shows a real Suminagashi design which is probably created in a fluid with low viscosity. Note that, except for the vortex sheet experiments discussed above, exact flow parameters or the initial and boundary conditions are not known for any of the real examples. Numerical solutions are based on my guesses at these properties.

2. Turbulence Around a Moving Boundary: Small Scale

In the second experiment boundary point moves faster on the same predefined path. In order to keep the evolution of the fluctuations more local the maximum limit for the velocity difference in random force conditioning is slightly increased, $\theta = 0.03$; increasing this limit restrains the random force generation. Also to prevent overspreading of the vectors, momentum step is not increased, $momentum\_steps = 1$. In order to create small scale eddies averaging level is kept low, $smoothing\_steps = 2$. For more turbulent behaviours, viscosity (dissipation coefficient) is decreased ($\gamma = 0.01$) while the magnitude of the random force is increased, $\alpha = 0.35$ as well as the divergence level ($divergence\_steps = 12$). Figures 5.41 and5.42 show the highly turbulent flow with small scale eddies developing around the boundary more locally than the previous example. Obviously this is not a traditional marbling pattern; it is difficult to produce this type of turbulence in reality and even so it would be almost impossible to capture the patterns as the flow develops very fast. Numerical results are similar to real turbulence examples, see photographs of turbulence behind a jet in [157].

3. Surface Tension Effects Under Turbulence

In this experiment, large scale ($smoothing\_steps = 6$, $\alpha = 0.7$), turbulent ($\gamma = 0.02$) flow with relatively high level of incompressibility ($divergence\_steps = 10$) is triggered by the interface. Conditioning for the fluctuations is defined as $\alpha = \alpha(\rho_1 - \rho_2)$. Since the effect of fluctuations is exaggerated and turbulence is large scale, turbulence characteristics become almost homogeneous

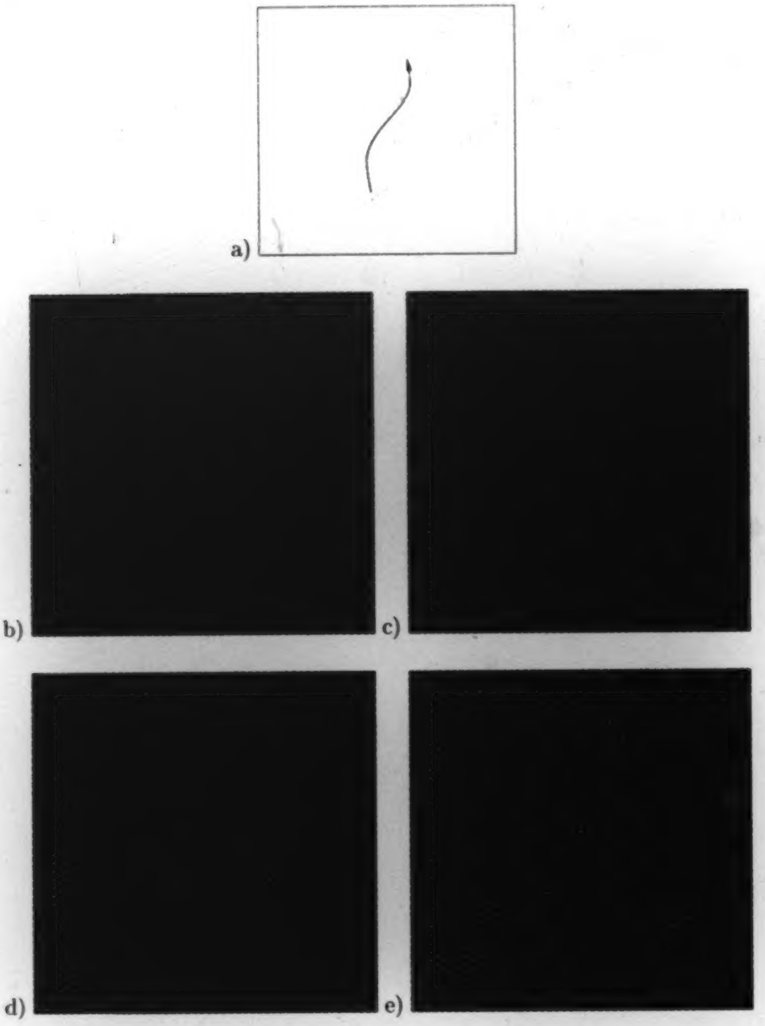
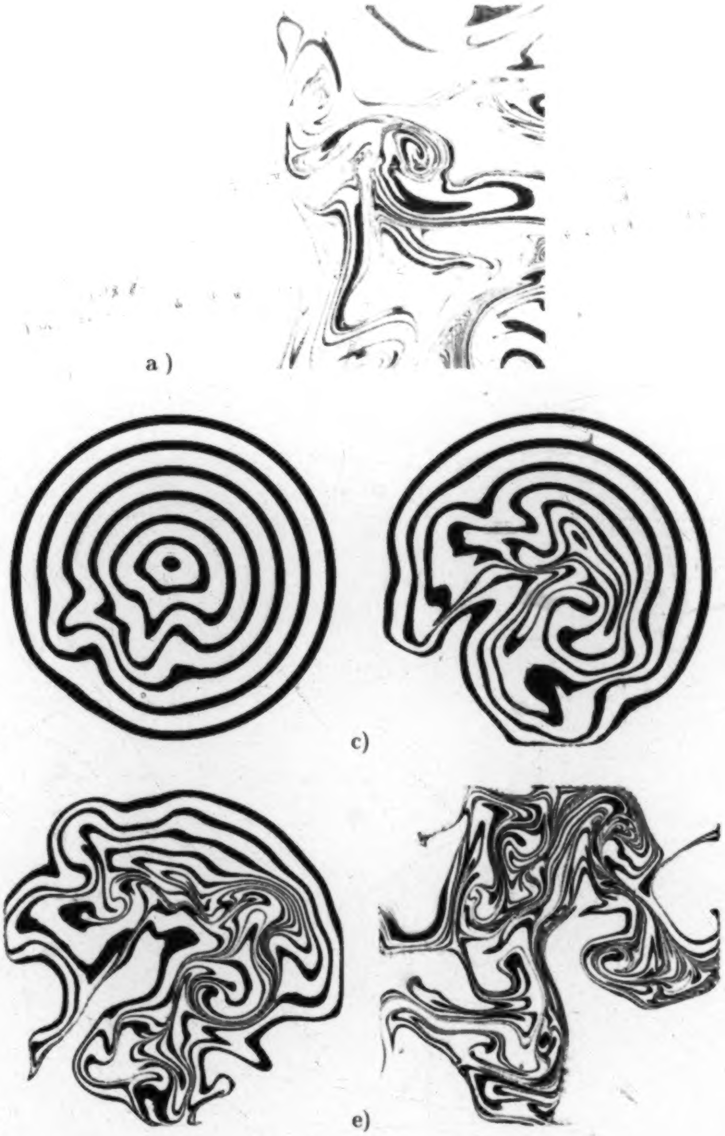


Figure 5.39: Large scale turbulence around the boundary: a) Boundary path (approximate), b-e) Velocity vectors, c)41 Δt, d)101 Δt, e)171 Δt.


a)

b) c)

d) e)

Figure 5.40: Large scale turbulence around the boundary: a) Traditional Suminagashi design, b-e) Advection patterns, b)40 Δt, c)100 Δt, d)170 Δt, e)270 Δt.

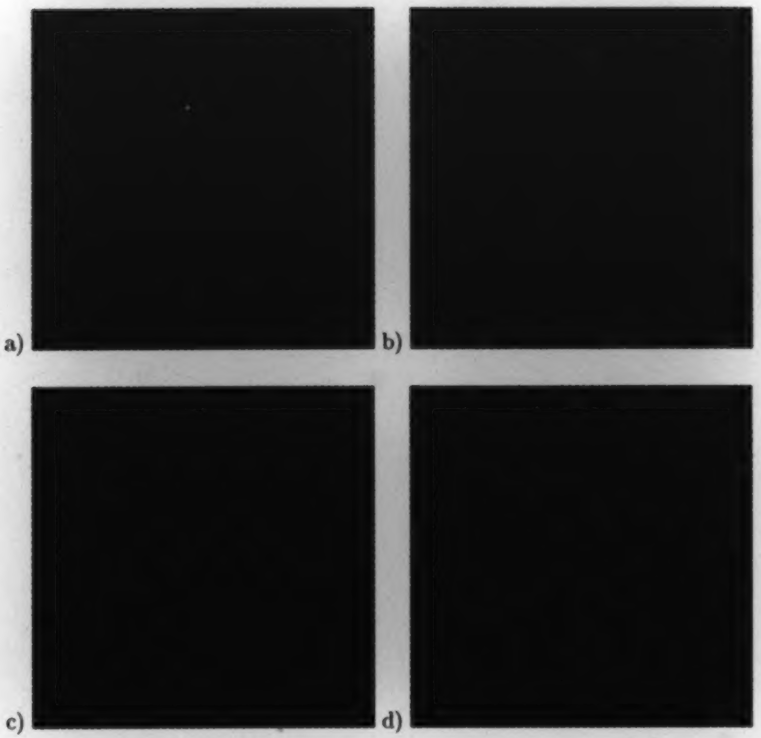


Figure 5.41: Small scale turbulence around the boundary: a-d) Velocity vectors, a)31 $\triangle t$, b)71 $\triangle t$, c)151 $\triangle t$, d)161 $\triangle t$.

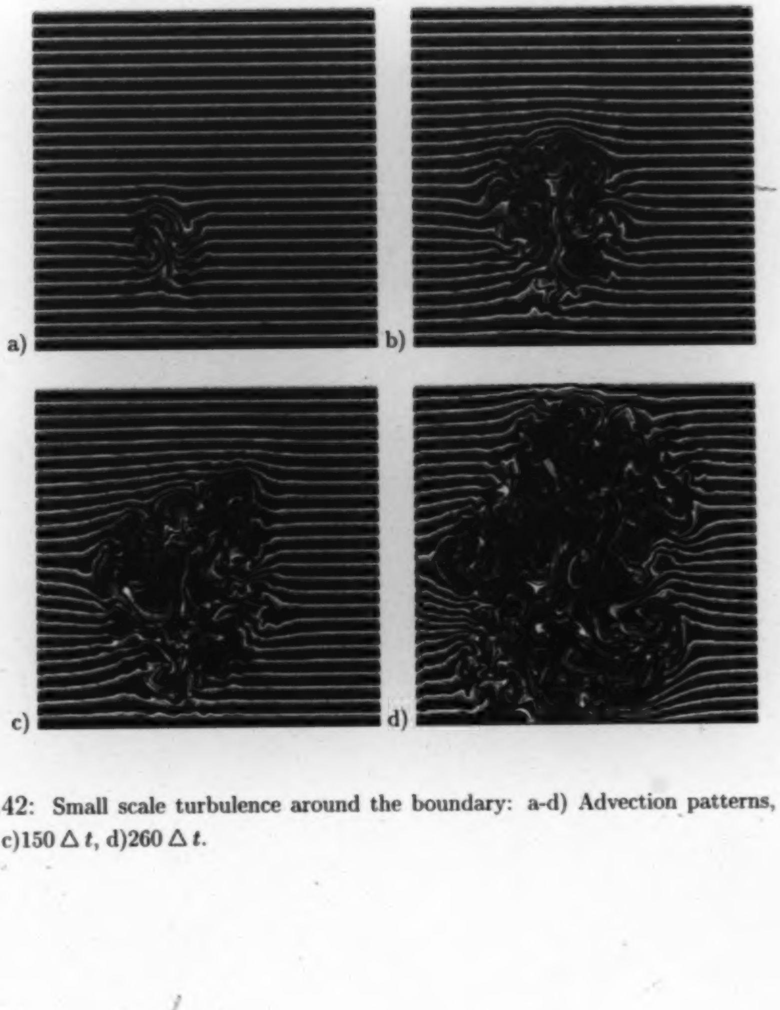


Figure 5.42: **Small scale turbulence around the boundary: a-d) Advection patterns, a)30 Δt, b)110 Δt, c)150 Δt, d)260 Δt.**

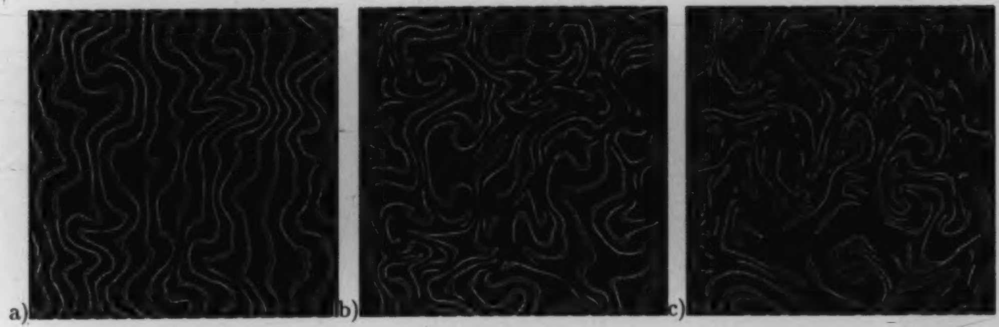


Figure 5.43: Surface tension effects under large scale turbulence triggered by the interface: a) 33 Δt, b)80 Δt, d)170 Δt.

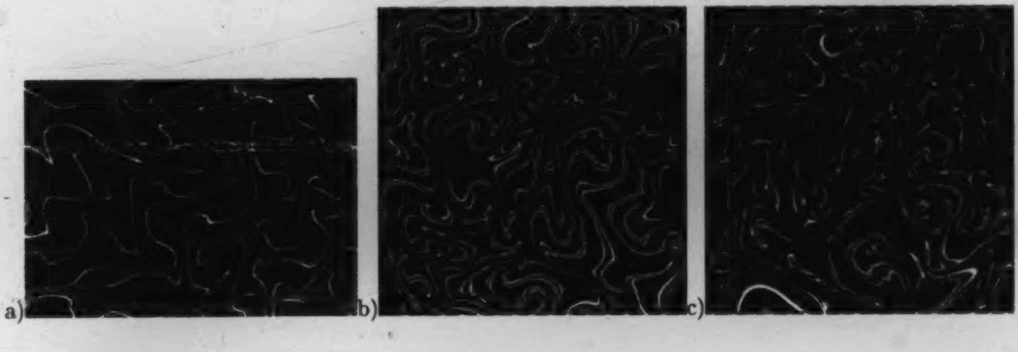


Figure 5.44: Same interface triggered turbulence (as in 5.43) but without the surface tension effects.a) real marbling design b)85 Δt, c)170 Δt.

after a while. In this experiment my goal was to to demonstrate the surface tension effect in a turbulent flow. Interface is described by the Cahn-Hilliard equation (with $\hat{\xi} = 0.001$). However unlike the interface experiments in above *nonlocal effects* are not applied. As a result interface is thinner but coalescence is slower. Density parameter represents two fluids ($\rho = 0$, $\rho = 1$). Two colors are mixed according to this parameter in a way to emphasize the interface. Easily broken lines can be observed in Figure 5.43. At advanced stages of the flow, coalescence also takes place. For comparison, same experiment is repeated without the surface tension effect. The image is represented in RGB format but a density parameter which is subject to Cahn-Hilliard diffusion is also advected in addition to the RGB components to obtain the same flow properties as above which are based on interface conditioning. In Figure 5.44, we observe that the lines stretch without breaking for a longer duration in this case. Note that, patterns created in this way are similar to some of the traditional marbling examples (Figure 5.44a).

4. Fractured Velocity Fields

Here my goal was to obtain the fractured velocity patterns in real marbling examples in Figures 5.45a, and 5.46a These are conventional Suminagashi designs, which are usually created by fanning over the fluids with low viscosity. Fluids are usually prepared by dropping dilute paint solutions on a background liquid which is only water without any additives. Paints easily oscillate forming broken irregular patterns by chaotic forces of the wind. I used a rotating mixing device (as described in chaotic mixing experiments), but with multiple pins and a bigger radius to simulate a fan like comb. In order to create the fractured oscillation in the velocity, first the relative velocity based condition limit for the fluctuations is increased ($\theta = 0.1$). When this value is high, velocity conditioning produces an irregular fluctuating motion only around high velocity oscillations which are mostly near the boundary since dissipation becomes more powerful on small scale velocity fields farther from the boundary. And when θ limit is low this effect is less prominent and more independent velocity fluctuations are observed with more vortical motion as seen in the previous examples. In order to prevent the vorticity creation, but also to keep the flow incompressible at the same time the flow parameters are defined as follows. First the smoothing level is adjusted (*smoothing_steps* = 2) low enough not to smooth out the broken velocity structures, but high enough to prevent the creation of small scale vortices. Boundary rotates in one direction and usually applied in parallel lines instead of overlapping rotations. And the rest of the flow parameters are defined as follows ($\gamma = 0.01$, $\alpha = 0.7$,*divergence_steps* = 1). Figure 5.45 shows the flow patterns obtained in this way with a comparison to a real Suminagashi design. In order to obtain less vortical flow with coarser fluctuations, I increased the random force magnitude ($\alpha = 1.0$) and the smoothing level (*smoothing_steps* = 3). In Figure 5.46 we see the flow with more fractured patterns.

5. Smooth Oscillations

In some Suminagashi techniques, dispersant agent is added to some of the paints during the flow; this creates a vibrant motion in the paints (Figure 5.48c) due to density differences or other fluctuations in the pigmented solutions. In order to obtain a similar flow I used interface based conditioning for the random force. Random oscillations are relatively high ($\alpha = 0.2$) in a low viscosity fluid ($\gamma = 0.05$). In order to create a slightly vibrant or fluctuating, incompressible motion smoothing level is increased (*smoothing_steps* = 6) and divergence level is kept moderate (*divergence_steps* = 5). Figures 5.47, 5.48 show the fluid motion which is only due to interface fluctuations, without the influence of an external boundary force. (Note that, at later stages oscillations are exaggerated.)

5.2.4 Density Fluctuations, Turbulent (Eddy) Diffusion.

Transport of particles in a fluid due to diffusion is a complex phenomena. Basic diffusion involves the flow in the direction of decreasing concentration gradients. According to Einstein's theory, averaging the square distance [1] taken by the randomly moving particles suspended in a liquid gives the molecular diffusion equation:

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x_i^2}, \tag{5.36}$$

---

[1] Linear averaging the distance traveled by a Brownian particle gives zero.

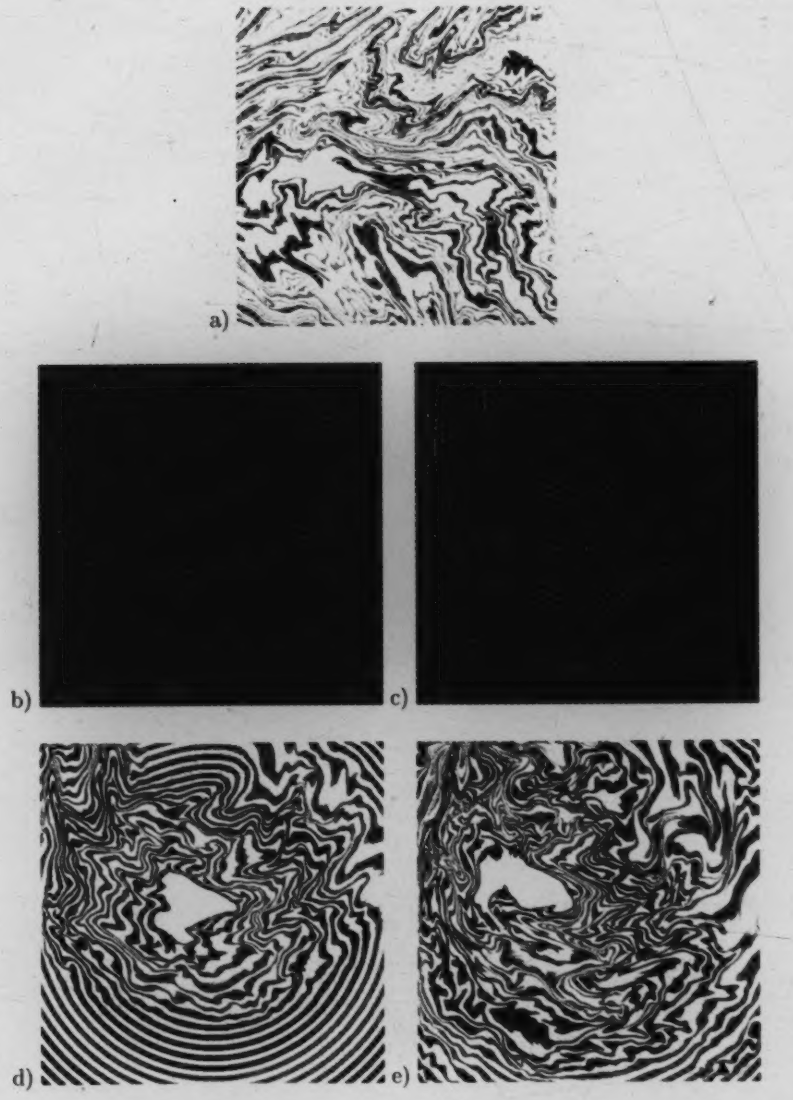


Figure 5.45: Fractured velocity fields: a) real marbling design , b-c) velocity vectors from the numerical simulation, d-e) advection patterns, d)20 $\triangle t$, e)40 $\triangle t$.

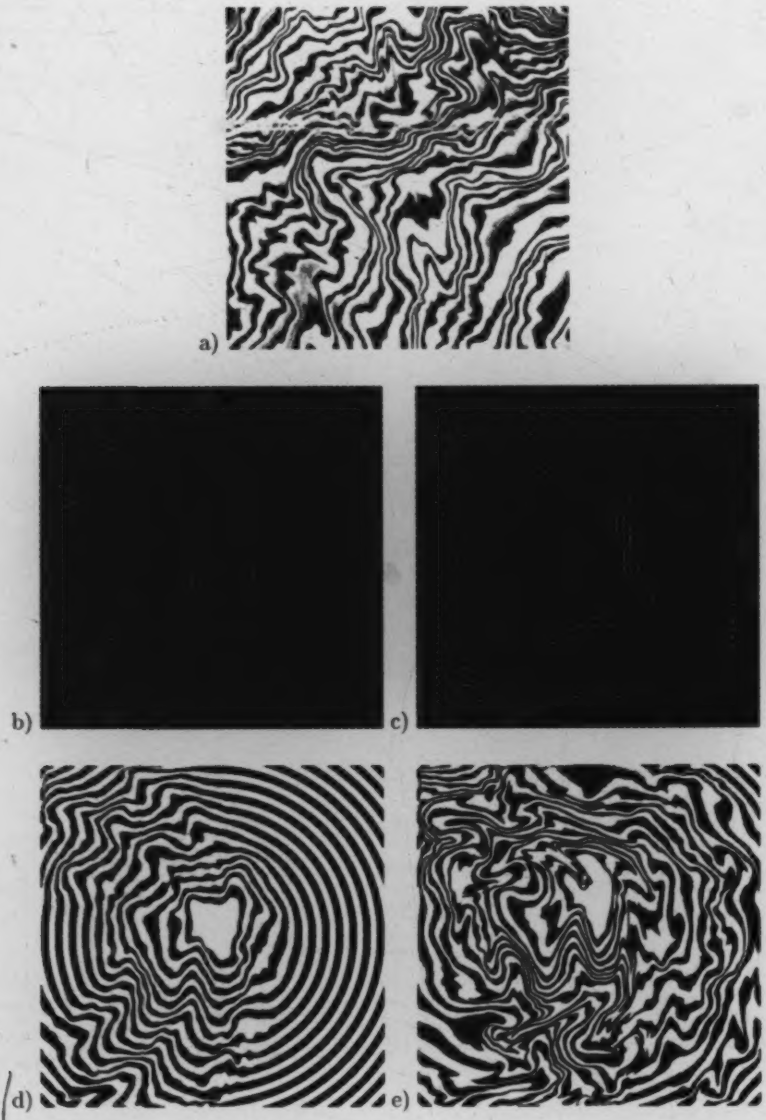


Figure 5.46: Fractured velocity fields (increased fluctuations). a) Real marbling design, b-c)velocity vectors from the numerical simulation, d-e)Advection patterns, d)10 $\triangle t$, e)30 $\triangle t$

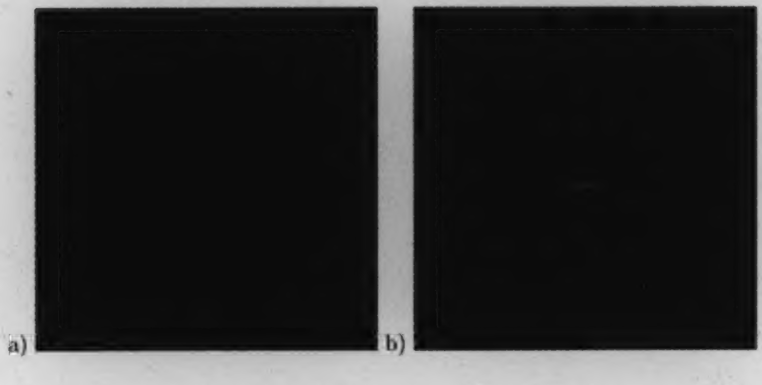


Figure 5.47: Velocity fields with smoother oscillations: a-b) Velocity vectors from the numerical simulation.

where diffusion coefficient, D, is constant and a property of the fluid and the suspended solution. This gradient based diffusion is the most common form of diffusion. Molecular diffusion has not been included in most of the marbling experiments since the molecular diffusivity of most marbling paints is very low (as mentioned before). However, eddy diffusion (or turbulent diffusion) probably plays an effective role in many marbling flows. Eddy diffusion refers to the concentration fluctuations caused by small scale eddies in a turbulent flow. In other words, small scale eddies cause the particles move randomly and accelerate diffusion. In classical turbulence modeling, in order to describe the effect of fluctuations, transport equation is represented in terms of concentration and velocity variables decomposed into average and fluctuating components. Let, $(U + U')$ and $(C + C')$ be the velocity and concentration values represented in this way respectively. When these are substituted into the transport (advection-diffusion) equation, it takes the form [128],

$$\frac{\partial C}{\partial t} = -U_i \frac{\partial C}{\partial x_i} + D \frac{\partial^2 C}{\partial x_i^2} - \frac{\partial}{\partial x_i} \overline{U_i' C'}, \tag{5.37}$$

where the overbar denotes averaging. The last term on the right hand side indicates the extra term arising due to turbulent fluctuations which of course raises a closure problem. Since molecular diffusion is very small compared to the dispersion caused by fluctuations it is usually omitted. A common approach assumes that the transport of the fluctuating component is related to the concentration gradient and describes the extra term in terms of eddy diffusivity as follows,

$$\overline{U_i' C'} = -D_{\varepsilon_i} \frac{\partial C}{\partial x_i}. \tag{5.38}$$

Several studies have been done to develop closure models that describe the diffusion coefficient, D_ε, some of which are based on based on random walk calculations [128, 107].

I should also mention the well-known power scale law, derived by Batchelor (and Richardson), describing the relation between the diffusion coefficient and the average distance between the particles, r, as follows,

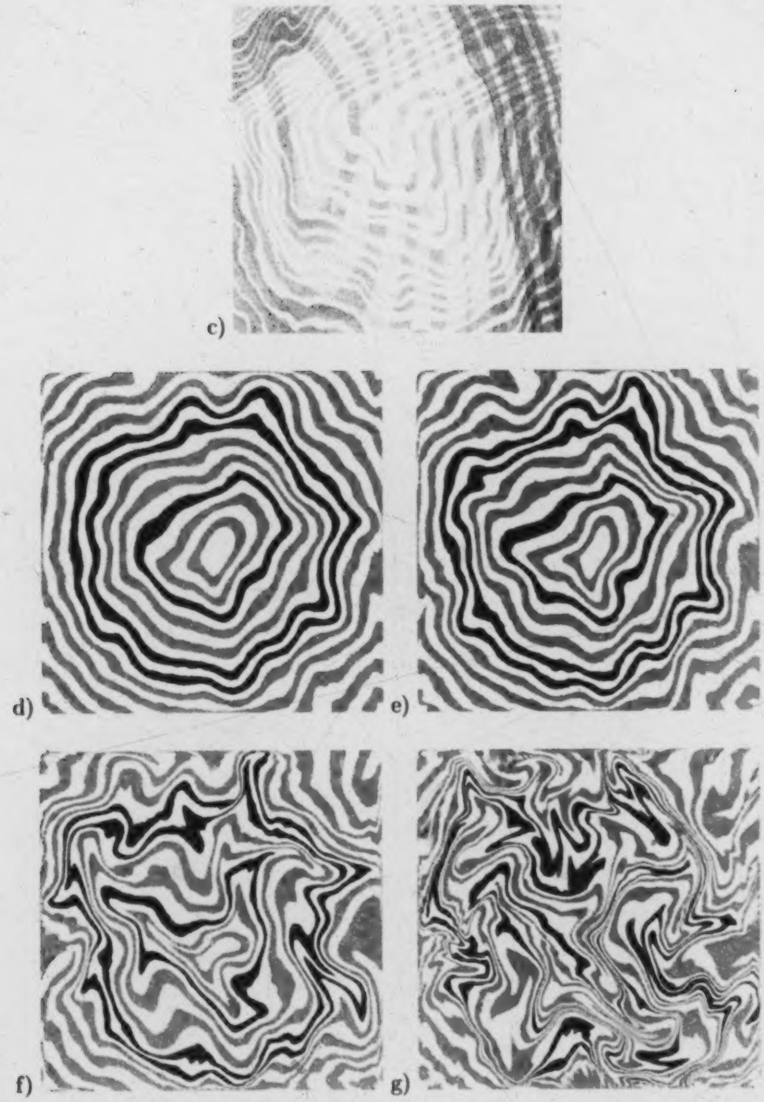


Figure 5.48: Velocity fields with smoother oscillations: a) Real marbling design , d-g) Advection patterns from the numerical simulation, d)$40 \triangle t$, e)$60 \triangle t$, f)$120 \triangle t$, g)$240 \triangle t$.

$$D(r) \sim r^{4/3}. \tag{5.39}$$

This relation states that, a material expands continuously under turbulent diffusion; in real life molecular diffusion usually slows down this process.

5.2.4.1 Density Fluctuations in Digital Marbling

Instead of following the more systematic traditional approaches, I propose a rather unconventional but practical solution to this problem where the user have flexibility in creating different effects. This system also makes use of the underlying mesoscale fluid model. In order to simulate the concentration fluctuations, a reaction term is included in the advection equation, which depends on the history of the particle trajectories. Note that, a similar idea has been used in computational physics [118], to study fractal pattern formation in chemically reacting flows under chaotic advection.

Investigating the Lagrangian motion of randomly moving particles is one of the ways to study turbulent diffusion [107]. Considering that the turbulent diffusion is provoked by mesoscale eddies and given the semi-Lagrangian characteristic of the advection solution used in the marbling model, the random motion of particles can easily be reflected to the concentration evolution as fluctuations. Based on these, the reaction term is defined as follows:

$$R(C) = F(\mathbf{x}^*), \tag{5.40}$$

where $\mathbf{x}^*$ represents the position backtracked in time (recall that this is already defined in the advection discretization). And F denotes a smoothing function which relates the two dimensional position vector to a scalar value. I used the sinusoidal function to control the oscillation in the range $[-1, 1]$ as follows,

$$F(C, \mathbf{x}) = A \sin(2\pi f\, x^*) \sin(2\pi f\, y^*), \quad \mathbf{x} = (x, y), \tag{5.41}$$

where A is a constant used for scaling and f is a frequency component used to determine the frequency of the oscillations in space. With this function, oscillations in the concentration are directly connected to the fluctuations in the velocity; concentration fluctuations are turbulent if the flow is highly turbulent, and they are smoother when the flow is more viscous. Figure 5.49 shows the density fluctuations for both small and larger scale velocity fluctuations. Velocity fluctuations are triggered by the interface of a ring located at the center. Then averaging is applied at two scales. Note that, in these figures material advection is not applied, to demonstrate only density fluctuations. Figure 5.50 demonstrates the role of frequency, f. Image is advected in a steady flow on frequency is changed in three levels. We see how oscillations increase as frequency increases from top to bottom.

5.2.4.1.1 Experimental Results

I have performed two experiments to demonstrate the effect of fluctuations in a turbulent and a relatively smoother flow.

1. In the first experiment, turbulent mixing of two fluids, $\rho = 0$, $\rho = 1$, is simulated. Fluids have a high density (or temperature) difference; fluctuations are triggered by the density difference

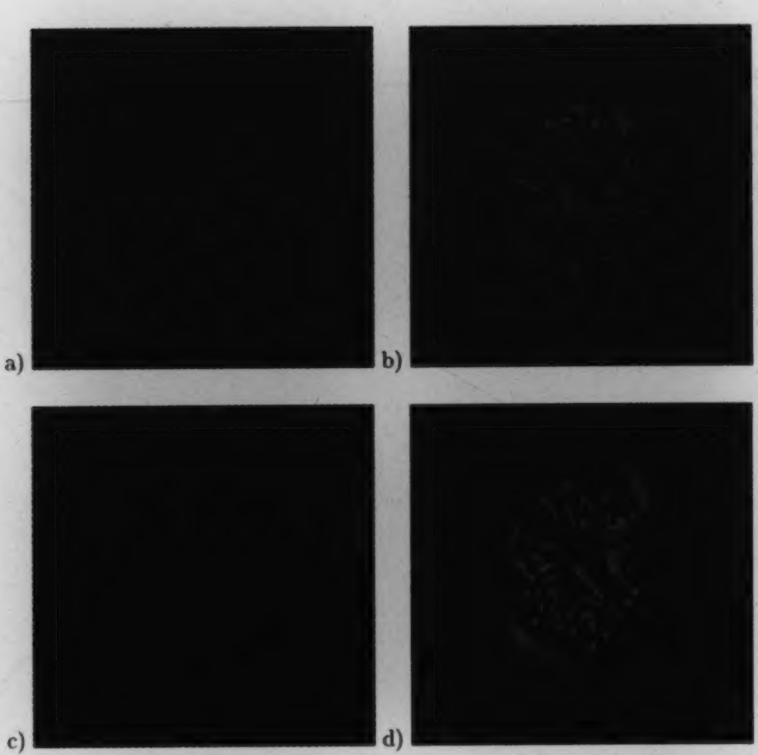


Figure 5.49: Velocity vectors and the corresponding density fluctuations, for flows with small scale fluctuations (a,b) and and smoother vectors (c,d).

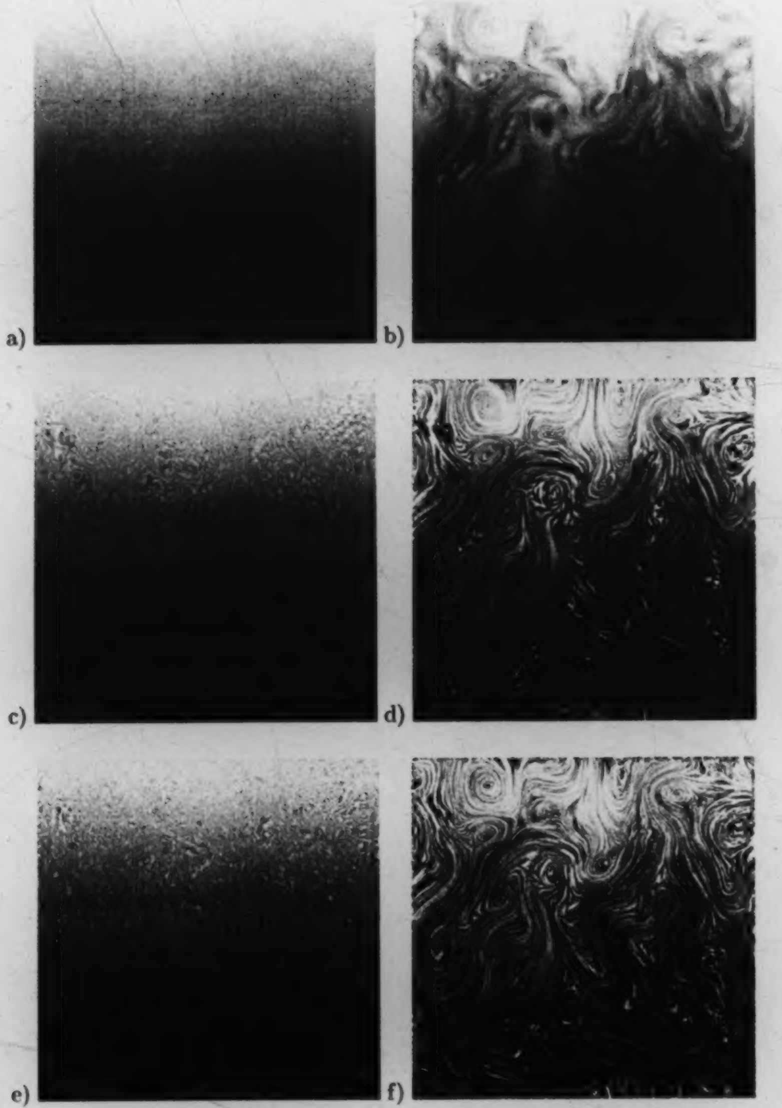


Figure 5.50: Advection-reaction with increasing density fluctuation frequency: a,b) f=1, c,d) f=2?, e,f) f=3?, a,c,e) 5 Δt , b,d,f) 35 Δt.

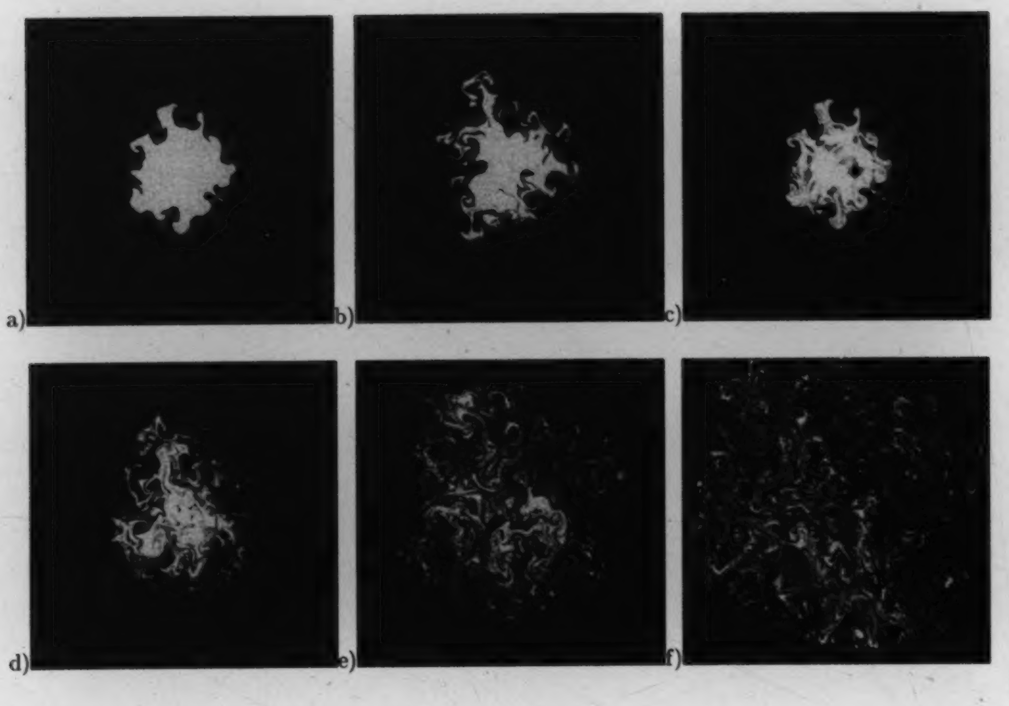


Figure 5.51: Turbulent diffusion: a,b) Advection without the reaction term, c,f) Advection with density oscillations, a,c) 20 Δt , b,d) 60 Δt, e) 132 Δt , f) 211 Δt.

between the fluids. The condition for the random force generation is given by,

$$\alpha_{ij} = \alpha(\rho(\mathbf{x}_i) - \rho(\mathbf{x}_j)). \tag{5.42}$$

Here i and j correspond to the indices of two adjacent Voronoi cells. Fluid density is obtained from the image grid position corresponding to the center of the Voronoi cell; thus it is assumed that the center point value represents the average density of the surrounding Voronoi area. Reaction term is activated only if $\rho > 0.005$, this causes the reaction to evolve around the interface, and stronger in one fluid, it also controls its evolution speed. In the fluid model, small scale fluctuations are used ($\alpha = 0.1\, \gamma = 0.1$, $smoothing\_steps = 1$), with a higher divergence field ($divergence\_steps = 5$). In rendering, density parameter is simply used for blending two colors:

$$color = color_1 \rho + color_2(1 - \rho). \tag{5.43}$$

Figure 5.51 shows the frames from the evolution of the density field. This method can also be used for the simulation of other phenomena such as combustion and smoke.

2. In the second experiment flow is defined similarly in fractured velocity experiments, with the parameters ($smoothing\_steps = 3$, $divergence\_steps = 1$, $\gamma = 0.1$, $\alpha = 0.5$, $\theta = 0.05$) to

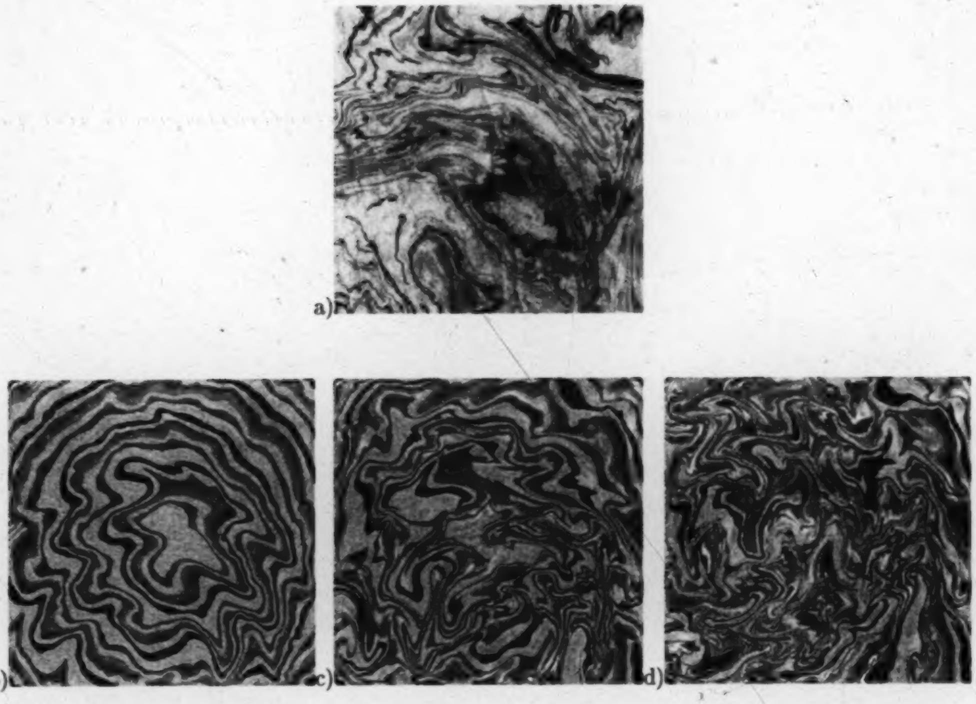


Figure 5.52: Smoother density oscillations in a flow with fractured velocity fields: a) Traditional marbling design, b-d) Numerical solution, b) 15 Δt , c) 45 Δt, d) 90 Δt.

simulate smoother fluctuations in the density. Figure 5.52a shows a real marbling example which probably has been influenced by density fluctuations and Figure 5.52b-d shows the numerical results. Image is represented by RGB components and an extra density field which is subject to reaction. Color components are expressed as,

$$pixel(i,j)_{R,G,B} = pixel(i,j)_{R,G,B} * \rho. \tag{5.44}$$

A scaling factor, $A = 0.5$ is used and density is adjusted as below to control its range.

$$\rho = (\rho + 1.3)/2.6 \tag{5.45}$$

Note that, to be able to match the original oscillations fluctuations are slightly smoothed using diffusion in this experiment. (Diffusion equation is solved with a spectral method similarly in Cahn-Hilliard equation.)

As seen in the experimental results, this is a very practical solution for modeling different effects, considering that it is readily built on the current advection solution. It is also effective in the sense that it makes use of the mesoscopic properties of the fluid model. In the future, it would be interesting to study physical (and statistical) properties of eddy diffusion simulation.

Chapter 6

Conclusion

In this dissertation, inspired by the complexity and variety of traditional marbling techniques, I have addressed the problem of using certain aspects of fluid dynamics as an artistic tool for computer graphics. In this context, this thesis addresses various problems in numerical fluid modeling. This chapter presents a summary of the methods proposed and a discussion on the current results and future research directions.

6.1 Summary

Traditional marbling, in its various styles covers a wide range of topics in fluid dynamics. It involves very complex physical and chemical processes some of which does not even have a theoretical description. Although it may not be possible to model all its details with the current knowledge and technology, since complete physical accuracy is not our main concern in computer graphics, especially in simulations for artistic purposes, this thesis aims to provide a similarly complex system, using computational methods. Considering the physical aspects of traditional marbling as well as the numerical requirements for certain problems in fluid modeling this thesis addresses the digital marbling problem mainly in three categories: modeling viscous fluid equations, vortex sheet dynamics and transport equations.

Viscous fluid model is based on stochastic mesoscale fluid dynamics. My goal was to use small scale mesoscopic details to obtain complex fluid behaviours at the macroscopic level while exploiting the flexibility of the random characteristics of the mesoscopic forces in creating different effects. To achieve this goal I developed a multiscale fluid model based on filtering the fluid equations using a kriging averaging method. Without the fluctuations the method reduces to a basic Navier-Stokes solution which can be used to model relatively low Reynolds numbers flows. Properties of the mesoscale fluctuations are altered to control the scale of fluctuations and how they influence different scales. In general, there are two important features of this multiscale fluid model particularly from the viewpoint of computer graphics. First, one can simulate a wide range of complex, unpredictable flow behaviours ranging from smooth oscillatory behaviours to highly turbulent flows with the same model. Second, this system allows user control at every scale of the flow.

Vortex sheet model is based on a solution of inviscid equations in which the vortex sheet properties are represented by a smoothed level set parameter. One significance of this model is that it allows the simulation of complex vortex sheet patterns with a very simple and practical computational method. Another important aspect is the suitability of the level set properties for further improvisation. Based on these, the level set model is introduced as a general tool for both realistic simulation of vortex sheet dynamics, certain aspects of vorticity dynamics and 2D turbulence, as well as for creating other artistic effects such as the traditional comb patterns.

In the transport model, semi-Lagrangian advection method is improved by the use of a high order spatial interpolation. In this way, level of detail was increased in the advected patterns while the diffusive behaviour was reduced. I also introduced a sharp interface method in the advection solution; this method was used to obtain similarly sharp, even jagged edges in advection lines, which are a typical characteristic of some marbling designs. Furthermore two reaction models are introduced. One is used to simulate material expansion under convection and the other models density fluctuations. In addition, in order to model interface dynamics a diffuse-interface approach is introduced. This method which describes the interface evolution through Cahn-Hilliard equation is based on interface physics.

The results are classified according to flow properties. Properties of the advection, reaction and interface methods are demonstrated in viscous flows, where the mesoscopic fluctuations are discarded, under the investigation of chaotic mixing. High level of detail due to 5th order B-splines were evident in the advected patterns. Sharp interface lines and chaotic patterns were comparable to real marbling flows. The effect of reaction term and the interfacial events, although not necessarily similar to the traditional techniques, are demonstrated under different mixing protocols.

In the multiscale model, I have performed several experiments to produce fluctuating flows at different scales and complexities. In these flows fluctuations triggered by velocity and density differences are used. By adjusting the control parameters of the multiscale model the following flows are created:

- Turbulent motion caused by moving boundaries at small and larger scales.
- Turbulent motion caused by evolving interfaces including the surface tension effects.
- Two types of fluctuating flows at different scales; one with highly fractured velocity fields and the other with smooth oscillations at fluid interfaces.

Furthermore, density fluctuations are demonstrated at two different scales:

- Turbulent diffusion in a highly turbulent flow.
- Smooth density oscillations in a flow with smoother velocity oscillations.

The results were compared to various Suminagashi techniques as well as to other highly turbulent real flows.

In the vortex sheet model, properties of the level set function and the numerical method are used to simulate the following realistic flows:

- Vortex sheet patterns created by different types of boundaries. These are compared to my experiments with the real fluids on a one-to-one basis and also to soap film experiments in the literature.

- Vortex sheet interaction and two dimensional grid turbulence which are comparable to both soap film experiments and numerical solutions.

- Evolution of secondary vortices in vortex sheets at two different levels of strength.

Finally, further improvisation of the level set properties are demonstrated by creating two well-known comb patterns following the same instructions in real marbling examples.

In conclusion, in all parts of the model, I have compared the results either with the traditional marbling examples or with real fluid experiments. Or in other cases, where certain properties of the fluid methods were used to create abstract designs, I have demonstrated the mathematical aspects of the model. In this way, I have covered most marbling styles except for the ones which make use of certain chemical reactions. I didn't have much difficulty in replicating the traditional counterparts, even for the most unconventional ones. Aside from this, also considering my experiments with the real technique, I found the computer version to be a more versatile environment both in terms of creativity and practicality. In this sense, I believe I have achieved my goal in developing a new computer tool which, by taking advantage of the computational methods, provides a powerful fluid dynamics simulation environment for artistic purposes.

6.2 Future Work

6.2.1 Experiments

For some parts of the fluid model, although the corresponding numerical solution has been evaluated with test cases, it hasn't been demonstrated in marbling experiments.

- Although relatively small scale surface tension effects merely based on Cahn-Hilliard equation have been demonstrated, Navier-Stokes-Cahn-Hilliard coupling has been neglected in the experiments mainly due to the computational inefficiency of this solution. In order to improve the computation of the surface tension force in the momentum equation, a more accurate approximation on a higher resolution grid can be considered. And this part of the surface tension model can be used to simulate bubble dynamics.

- In simulation of density fluctuations, the reference image I used was somewhat ambiguous regarding its fluctuation properties. A more characteristic marbling example can be found where the eddy diffusion influence is obvious, to use as a reference in comparison with the numerical results.

6.2.2 Further improvements

Concerning the accuracy of the fluid model, there is actually no limit to further improvements as the numerical methods are still far from approximating the continuous fluid equations with full accuracy. However in the marbling model, since high Reynolds number flows are handled with alternative approaches I don't find it necessary to increase the accuracy of the discretization any further. Computational efficiency of the numerical model on the other hand can be improved as follows:

- Finite volume discretization of the Voronoi fluid model is suitable for parallelization. In order to speed up the evolution of flow vectors, parallel techniques can be considered both for momentum and incompressibility solutions.
- In the current system instabilities can easily be controlled with kriging smoothing. However for unconditional stability fully implicit schemes which allow parallel implementation can be investigated. Parallel implementation is emphasized since the implicit formulation on a Voronoi grid would be costly.

Furthermore, in the semi-Lagrangian advection model, multiresolution methods can be considered in order to further increase the accuracy of the spatial approximation, that is, to decrease the diffusive behaviour while obtaining higher level of detail. However since this would be computationally demanding, its suitability for parallel implementation should also be considered.

6.2.3 Extensions

- **User interface development.** A high level user interface for the marbling model which can be used by the artists, constitutes the highest priority in future extensions. Currently only a primitive interface is provided for the multiscale fluid model that allows the control of basic flow parameters. In a more advanced user interface, the user can also define the random force conditioning for instance. It would be also nice to offer a tool where he can design the boundary configurations. Allowing different interaction types with the fluid would be interesting, for example, this could be a system which is sensitive to boundary pressure or speed. In the vortex sheet model, It would be nice to have an environment where the user can design both the level set and the smoothing function, considering the range, periodicity, smoothness and sign transition properties. Level set paths could be drawn on the screen or they could be designed according to predefined functions. Various perturbations can be added in the free drawn paths. Real-time interaction with the level set data should also be considered. That is, real-time evolution of the fluid equations while the level set line is being drawn by the user would be interesting. This might require further investigation on signed distance transformation algorithms. Furthermore, in the transport model, especially the diffuse-interface parameters, such as the interface thickness, mobility and nonlocal effects should be included in the user interface.

- **Chemical reactions.** We have seen some of the marbling patterns such as vein patterns, which are created merely based on the chemical properties of the paints without any manipulation. Marbling artists are now exploring several other chemical properties. In order to simulate these effects reaction models can be developed. A realistic modeling of the elasticity of the marbling paints should also be considered. This might require high resolution viscoelastic fluid simulation. One can also consider using different energy functionals in the Cahn-Hilliard equation to simulate several properties of the paints.

- **Rendering.** Currently used traditional, additive and subtractive color synthesis techniques are appropriate for emitting and transmitting materials, but they are not sufficient for representing pigmented surfaces which should be specified in terms of their reflective and transmissive characteristics. For a more realistic rendering of the layered pigment surfaces in marbling, Kubelka-Munk pigment mixing model can be used. Kubelka-Munk theory describes the optical properties of translucent media which absorb and scatter light.

6.2.4 Future Research Directions

- **3D marbling.** One of the most interesting applications of this model in the future would be to develop a 3D environment where the user can interact with the 2D flow on the surface of 3D objects. Multiscale fluid model is already built on a Voronoi structure. Only it might be necessary to reconsider the stability issues and the convergence properties of the projection method as the Voronoi cell distribution will change. However adapting the transport model to 3D surfaces (that is irregular grids) may not be so easy. In advection solution irregular B-spline approximations can be considered. In diffuse-interface model, finite element methods could be one alternative for the discretization of the Cahn-Hilliard equation, as the currently used spectral method assumes uniform grid structure. Similarly, for the vortex sheet model, fast solutions of Poisson equation on irregular grids can be investigated.

- **Dynamical systems.** I have examined some properties of the flows which produce chaotic advection, from a dynamical systems perspective. Other dynamical systems can be explored to create interesting patterns, even though they might not be related to realistic flows. Controllability of dynamical systems is not obvious, at least not in the sense of realistic fluid mixing, however perturbation of chaotic systems can be investigated.

- **Theory.** Even though it may not be essential for graphics modeling, I would also like to study the physical aspects of the mesoscale averaging method in theory to define its properties in terms of the physical criteria, and more importantly to see if the turbulence model is amenable to the usual tests.

Bibliography

[1] Abid, M., Verga, A. Stability of a vortex sheet roll-up. Physics of Fluids, Vol. 14 (11), November 2002, pp. 3829–3834.

[2] Aref, H., Stirring by chaotic advection. J. Fluid Mech., 143 (1984), pp.1–21.

[3] Barton, I.E. Improved laminar predictions using a stabilized time-dependent SIMPLE scheme. International Journal for Numerical Methods in Fluids, Vol. 28 (1998), pp. 841–857.

[4] Batchelor, G.K., Computation of the energy spectrum in homogeneous two-dimensional turbulence. Phys. Fluids Suppl. II, 12(1969), pp. 233–239.

[5] Beatson, R.K., Light, W.A., Billings, S. Fast solution of the radial basis function interpolation equations: domain decomposition methods. SIAM J. Sci. Comput. Vol.22 (5), pp.1717–1740.

[6] Bell, J.B., Colella, P., Glaz, H.M. A second order projection method for the incompressible Navier–Stokes equations. J. Comp. Phys. 85 (1989), pp. 257–283.

[7] Boek, E.S., Coveney, P.V., Lekkerkerker, H.N.W., Schoot, P. Simulating the rheology of dense colloidal suspensions using DPD. Physical Review E, Vol.55(3), pp.3124–,1997.

[8] Boghosian, B.M., Coveney, P.V., Emerton, A.N. A lattice-gas model for microemulsions. Proc. Roy. Soc. London. Ser. A., 452 (1996), pp. 1221–1250.

[9] Botta, N., Hempel, D. A finite volume projection method for incompressible flows on triangular grids. Technical Report Series A, 110, Institut fur Angewandte Mathematik, Hamburg, 1996.

[10] Bowyer, A. Computing Dirichlet Tessellations. Computer Journal, 24(2), 1981, pp. 162–166.

[11] Boyland, P., Aref, H. Stemler, M. Topological fluid mechanics of stirring. J.Fluid.Mech., 403(2000), pp.277–304.

[12] Bowyer, A. Computing Dirichlet Tessellations. Computer Journal, 24(2), 1981, pp. 162–166.

[13] Brackbill, J.U., Kothe, D.B., Zemach, C. A continuum method for modeling surface tension. Journal of Computational Physics, Vol. 100 (1992), pp. 335–354.

[14] Bunch, J.R., Kaufman, L. Some stable methods for calculating inertia and solving symmetric linear systems, Math. Comp., 31 (1977), pp.163–179.

[15] Cahn J.W., and Hilliard, J.E., Free energy of a nonuniform system, Interfacial free energy. J. Chem. Phys., 28 (1958), pp.258–267.

[16] Celik, I. Introductory Turbulence Modeling. West Virginia University, Mechanical and Aerospace Engineering Department. 1999.

[17] Chambers A., Suminagashi: The Japanese Art of Marbling,Thames and Hudson, 1991.

[18] Chen, J., and Lobo, N. Toward interactive-rate simulation of fluids with moving obstacles using Navier–Stokes equations. Graphical Models and Image Processing, 57(2):107–116,1995.

[19] Chen, C.K., and Fife, P.C. Nonlocal models of phase transitions in solids. Adv. Math. Sci. Appl. 10 (2000), 821–849.

[20] Chorin, A.J., A Numerical solution for solving incompressible viscous flow problems. J. Comp. Physics, 2 (1967), pp. 12–26.

[21] Chorin,A.J. Numerical simulation of the Navier–Stokes equations. Mathematics of Computation, 22(104), 1968, pp.745–762.

[22] Chorin, A.J. Numerical study of slightly viscous flows. J. Fluid Mech., 57 (1973), pp. 785–796.

[23] Christiansen, J.P. Numerical simulation by the method of point vortices. J. Comput. Physics., 13 (1973), pp.363–379.

[24] Clercx, H.J.H., Maassen S.R., van Heijist, G.J.F. Decaying two-dimensional turbulence in square containers with no-slip or stress-free boundaries. Physics of Fluids, Vol. 11(3), March 1999.

[25] Couder, Y., Chomaz, J.M., Rabaud, M. On the hydrodynamics of soap films. Physica D. Vol. 37(1989), pp.384–405.

[26] Couder, Y., Basdevant, C. Experimental and numerical study of vortex couples in two-dimensional flows. J. Fluid Mech., Vol. 173 (1986) , pp. 225–251.

[27] Cressie, N. Statistics for Spatial Data. John Wiley and Sons, New York, 1991.

[28] Cummins, S.J., Rudman, M.An SPH Projection Method. Journal of Computational Physics, Volume 152 (2) ,1989, pp. 584–607.

[29] Curtis, C.J., Anderson, S.E., Seims, J.E., Fleischer, K.W., and Salesin, D. Computer generated watercolor. Proceedings of SIGGRAPH'97, August, pp. 421–430.

[30] Date, A.W. A complete pressure correction algorithm for solution of incompressible Navier–Stokes equations on non-staggered grids. Numerical Heat Transfer B., 29 (1996), pp. 441–458.

[31] Desbrun, M. Cani-Gascuel, M.P. Active implicit surface animation. Graphics Interface, 1998, pp. 143–150.

[32] Dodgson, N.A. Image resampling. Technical Report No. 261. Computer Laboratory, University of Cambridge. August 1992.

[33] Durikovic,R. Animation of Soap Bubble Dynamics, Cluster Formation and Collision. Computer Graphics Forum (Eurographics'2001 Proc.) 20(3), 2001, pp.67-76.

[34] Dzwinel, W., Yuen, D.A. Mixing driven by Rayleigh-Taylor instability in the mesoscale modeled with dissipative particle dynamics. International Journal of Modern Physics C. Vol. 12(1). 2001, pp. 91–118.

[35] Dzwinel, W., Yuen, D.A. Matching macroscopic properties of binary fluids to the interactions of dissipative particle dynamics. International Journal of Modern Physics C. Vol. 11(1). 2000. pp. 1–25.

[36] E, W., Palffy-Muhoray, P. Phase separation in incompressible systems. Physical Review E. Vol. 55 (4), April 1997, pp.R3844–R3846.

[37] Enright, D., Marschner, S. and Fedkiw, R., "Animation and Rendering of Complex Water Surfaces", SIGGRAPH 2002, ACM TOG 21, 2002, pp. 736–744.

[38] Espanol P. Fluid particle model. Physical Review E,57(3), 1998, pp. 2930–2948.

[39] Eyre, D.J. An unconditionally stable one-step scheme for gradient systems. Preprint. 1997.

[40] Eyre, D.J. Unconditionally gradient stable time marching in the Cahn-Hilliard equation. In computational and mathematical models of microstructural evolution. (San Fransisco, CA, 1998.) pp.39–46. MRS., Warrendale, PA, 1998.

[41] Fabritis, G.D., Coveney, P.V., Flekkoy, E.G. Multiscale modeling of complex fluids. Proceedings of 5th European SGI/Cray MPP Workshop, Bologna (Italy) (1999).

[42] Fattal, R. ,Lischinksi D. Target-driven smoke animation.SIGGRAPH 2004.

[43] Fedkiw R., Stam, J. and Jensen, H.W., Visual simulation of smoke, SIGGRAPH 2001, pp. 23–30 (2001).

[44] Feldman, B.E., O'Brien, J.F., Arikan, O. Animating suspended particle explosions. SIGGRAPH 2003. ACM Transactions on Graphics (TOG), 22(3), pp. 708–715.

[45] Fife, P.C. Models for phase separation and their mathematics. Electronic Journal of Differential Equations, Vol. 2000 (48) , 2000, pp.1–26.

[46] Flekkoy, E.G., Coveney, P.V., Fabritiis, G.D. Foundations of dissipative particle dynamics. Physical Review E,62(2), pp. 2140–2158, (2000).

[47] Fletcher, C.A.J. Computational techniques for Fluid Dynamics. Springer Verlag, Sydney, 1990.

[48] Fournier,P., Habibi,A.,and Poulin,P.Simulating the flow of liquid droplets. Graphics Interface, June,1998.

[49] Forster, D., Nelson, D.R., Stephen, M.J. Large-distance and long-time properties of a randomly stirred fluid. Physical Review A, Vol. 16(2), August 1977, pp.732–749.

[50] Foster, N.,and Metaxas D.. Realistic animation of liquids. Graphical Models and Image Processing,58(5), pp.471–483,1996.

[51] Foster, N.,and Metaxas D. Controlling fluid animation. Proceedings of CGI'97,1997.

[52] Foster, N.,and Metaxas D. Modeling the motion of a hot, turbulent Gas. ACM Computer Graphics, SIGGRAPH 97, August 1997.

[53] Foster, N. and Fedkiw, R., Practical animation of liquids. SIGGRAPH 2001, 15–22 (2001).

[54] Fractal Design Corp.Painter 5.0.

[55] Franke, R., and Nielsen, G. Smooth interpolation of larger sets of scattered data. Int. Journal for Numerical Methods in Engineering, 15, pp.1691–1704,1980.

[56] Fratzl, P., Penrose, O., Lebowitz, J.L. Modeling of phase separation in alloys with coherent elastic misfit. DIMACS Technical Report 99-42, August 1999.

[57] Friedlander, S.K. and Leonard Topper, L.Turbulence ; classic papers on statistical theory. Interscience Publishers Inc. New York. 1961.

[58] Frisch, U., Hasslacher, B., Pomeau, Y. Lattice-gas automata for the Navier–Stokes equation. Phys. Rev. Lett. 56 (1986), pp. 1505–1508.

[59] Frisch, U. Turbulence: The Legacy of A. N. Kolmogorov, Cambridge University Press, New York, NY, 1995.

[60] Fuhrmann, J., Langmach, H. Stability and existence of solutions of time-implicit finite volume schemes for viscous nonlinear conservation laws. 1998. Preprint no. 437.

[61] Gajewski, H. and Zacharias, K. On a nonlocal phase separation model. J. Math.Analy.Appl. 286 (2003), pp.11–31.

[62] Gamito,M.N.,Lopes,P.F., and Gomes,M.R. Two dimensional simulation of gaseous phenomena using vortex particles. In Proceedings of the 6th Eurographics Workshop on Computer Animation and Simulation, pp. 3-15. Springer-Verlag, 1995.

[63] Garcke, H., Preusser, T., Rumpf, M. A phase field model for continuous clustering on vector fields. IEEE Transactions on Visualization and Computer Graphics, Vol. 7(3), July-September, pp. 230–241, 2001.

C

[64] Gardiner, C.W. Handbook of Stochastic Methods. Springer, Berlin, 1985.

[65] de Gennes, P.G. Scaling Concepts in Polymer Physics (Cornell Univ. Press, Ithaca, 1979).

[66] Groot, R.D., Warren, P.B. Dissipative Particle Dynamics: Bridging the gap between atomistic and mesoscale simulation. Europhys. Lett, 42 (4), pp. 377–382 (1998).

[67] Grunebaum, G. Techniques for Marbleizing Paper. Dover Publications, Inc. New York. 1995.

[68] Gurtin M.E., Polignoe D,Vinals J. Two-phase binary fluids and immiscible fluids described by an order parameter. Math. Models Methods Appl. Sci., 6 (1996), pp.815–.

[69] Haeberli,P. Paint by numbers: Abstract image representations. Proceedings of SIGGRAPH'90, Computer Graphics, 24(4):215–223.

[70] Harabetian, E., Osher, S. , and Shu, C.-W. An Eulerian approach for vortex motion using a level set regularization procedure. J.Comput. Phys., 127 (1996), pp. 15–26.

[71] Hardy, R.L. Theory and applications of the multiquadric-biharmonic method. Computers Math. Applic. Vol. 19 (8/9), pp. 163–208, 1990.

[72] Hirt, C.W., Nichols, B.D. Volume of Fluid (VOF) method for the dynamics of free boundaries. J.Comput. Phys., 39 (1981), pp. 201–225.

[73] Hong,J.,and Kim,C., Animation of bubbles in liquid. Computer Graphics Forum (Eurographics'2003 Proc.) 22(3), pp.253–262 (2003).

[74] Hornby, P., Horowitz, F.G. Closely related alternatives to kriging interpolation. http://citeseer.ist.psu.edu/260847.html

[75] http://home.earthlink.net/~marutgers/. "Maarten Rutgers' web site."

[76] http://www.kenmusgrave.com. "Ken Musgrave"

[77] http://www.race.u-tokyo.ac.jp/%7Eyoichiro/index.html. "Yoichiro Kawaguchi- WEBSITE"

[78] http://www.cs.tut.fi/~agotchev/DIPII/lecture1.pdf.

[79] http://www5e.biglobe.ne.jp/~kuroda/mypagekaisetu-E.htm. "Suminagashi: Exposition."

[80] http://members.chello.at/ahmet.saral/menu.htm. "Ebru-A Turkish Art Going Back Five Centuries and the Art Works of Ahmet Saral."

[81] http://www.hollanders.com/marbles/french.htm. "French Marbles."

[82] http://chriscox.org/suminagashi/.

[83] http://www.ph.ed.ac.uk/~jmb/thesis/tot.html. "Lattice Boltzmann Methods in Interfacial Wave Modelling."

[84] http://www.small-expressions.com/ma.htm. "Galen Berry's Marble Art."

[85] http://swiss.csail.mit.edu/~jaffer/Geometry/Marbling. "Topological Computer Graphics."

[86] http://www.marbling.pe.kr/. "Suminagashi/ Marbling."

[87] Ierley, G.R., and Shkoller, S. Smoothing the wrinkles: a new model for surface tension, preprint, 2002.

[88] International Conference on Computer Graphics and Interactive Techniques archives. ACM SIGGRAPH Electronic art and animation catalogs.

[89] Jacqmin D. 1996. An energy approach to the continuum surface tension method. AIAA 96-0858. In Proceedings of the 34th Aerospace Sciences Meeting and Exhibit.

[90] Jacqmin D., Calculation of two-phase Navier–Stokes flows using phase-field modeling. Journal of Computational Physics, 155, pp.96–127. 1999.

[91] Jiang, G.S. and Peng, D. Weighted ENO schemes for Hamilton-Jacobi equations, SIAM J. Sci. Comput., 21, pp. 2126–2143, 2000.

[92] Kass M., and Miller, G. Rapid, stable fluid dynamics for computer graphics. Proceedings of SIGGRAPH'90, Computer Graphics, 24(3), pp. 49–57.

[93] Kellay, H., Goldburg, W.I. Two-dimensional turbulence: a review of some experiments. Reports on Progress in Physics. Rep. Prog. Phys. 65 (2002) pp.845–894. Institute of Physics Publishing.

[94] Kim, D., Choi, H. A second-order time-accurate finite volume method for unsteady incompressible flow on hybrid unstructured grids, Journal of computational Physics, vol. 162, pp. 411–428. 2000.

[95] Kim, J. and Moin, P. Application of a fractional-step method to incompressible Navier–Stokes equations. J. Comput. Phys. 59(1985), pp.308–323.

[96] Kirkpatrick, M.P., Armfield, S.W., Kent, J.H. Simulation of vortex shedding flows using high order fractional-step methods. ANZIAM J. Vol. 42(E), pp. C586–C876, 2000.

[97] Kojouharov, H.V., Welfert, B.D. A new numerical approach for the solution of nonlinear advection-reaction equations. IMACS 2000, International Conference on Scientific Computing and Mathematical Modeling.

[98] Kraichnan, R.H., Inertial ranges in two-dimensional turbulence, Physics of Fluids, 10(1967), pp.1417–1423.

[99] Krasny, R. D. Desingularization of periodic vortex sheet roll-up. Journal of Computational Physics. Volume 65 , Issue 2 (August 1986), pp. 292 – 313.

[100] Krasny, R. Vortex sheet computations: roll-up, wakes, separation. Lectures in applied mathematics. Vol. 28 (1991) American Mathematical Society, pp.385–401.

[101] Kueck,H.,Vogelgsan,C., and Greiner ,G. Simulation and rendering of liquid foams. Proceedings Graphics Interface, May,2002, pp. 81–88.

[102] Landau, L.D., Lifshitz, E.M. Statistical Physics: Course of Theoretical Physics, Vol. 5. (Pergamon, Oxford, 1980).

[103] Leo, P.H., Lowengrub, J.S., Jou, H.J. A diffuse interface model for microstructural evolution in elastically stressed solids. Acta Mater. Vol. 46, No. 6, pp.2113–2130, 1998.

[104] LeVeque, R.J. Numerical methods for conservation laws. Birkhauser, Basel, 1992.

[105] Li, S., Liu, W.K. Meshfree and particle methods and their applications. Applied Mechanics Review. 55(1), pp.1–34. 2002.

[106] Maassen, S.R., Clercx, H.J.H., van Hejist, G.J.F. Self organization of quasi-two-dimensional turbulence in stratified fluids in square and circular containers. Physics of Fluids. Vol. 14(7). pp. 2150–2168. July 2002.

[107] Majda, A.J., Kramer, P.R. Simplified models for turbulent diffusion: theory, numerical modeling and physical phenomena. Physics Reports. 314 (1999), pp. 237–574.

[108] Martys, N., Chen, H. Simulation of multicomponent fluids in complex three-dimensional geometries by the Lattice-Boltzmann method. Phys. Rev. E. 53 (1996), pp. 743–750.

[109] Maurer-Mathison D. The Ultimate Marbling Handbook: A Guide to Basic and Advanced Techniques for Marbling Paper and Fabric. Watson-Guptill Publishing (New York). 1999.

[110] Mavriplis DJ. Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes. AIAA J. 28(2), pp. 213–221. 1990 (February).

[111] Meier,B. Painterly rendering for animation. Proceedings of SIGGRAPH'96, pp. 447–484.

[112] Meijering, E.H.W., Zuiderveld, K.J., Vieger, M.A. Image reconstruction by convolution with symmetrical piecewise nth-order polynomial kernels. IEEE Transactions on Image Processing, Vol. 8(2), February, 1999. pp. 192–201.

[113] Miller, G. , and Pearce, A.Globular dynamics; a connected particle system for animating viscous fluids. Proceedings of SIGGRAPH'91, Computer Graphics, 23(3):305–309.

[114] Monaghan, J.J. Smoothed particle hydrodynamics. Annu. Rev. Astron. Astrophys. 30 (1992), pp.543–574.

[115] Monaghan, J.J. Simulating free surface flows with SPH. Journal of computational physics. 110 (1994), pp. 399–406.

[116] Muller, M. Mesoscopic and continuum models. In Encyclopedia of Physical Chemistry and Chemical Physics, Moore, J. H. ; Spencer, N. D. , Eds.; IOP: Bristol, 2001; Vol. II, pp 2087–2110.

[117] Nayroles, B., Touzot, G., Villon, P. Generalizing the finite element method: diffuse approximation and diffuse elements. Computational Mechanics. 10 (1992), pp. 307–318.

[118] Neufeld, Z., Lopez, C., Haynes, P.H. Smooth Filamental transition of active tracer fields stirred by chaotic advection. Physical Review Letters, Vol. 82 (12), March 1999, pp.2606–2609.

[119] Neufeld, Z., Haynes, P., Tel, T. Chaotic mixing induced transitions in reaction-diffusion systems. Chaos, Vol. 12 (2), June 2002, pp.426–438.

[120] Nguyen, D., Fedkiw, R. and Jensen, H., "Physically Based Modeling and Animation of Fire", SIGGRAPH 2002, ACM TOG 21, pp. 721–728 (2002).

[121] Novik, K.E., Coveney, P.V. Using dissipative particle dynamics to model binary immiscible fluids. International Journal of Modern Physics C. Vol. 8(4). 1997. pp. 909–918.

[122] Ottino, J.M. The Kinematics of Mixing: Stretching, Chaos and Transport. Cambridge University Press. Cambridge. 1989.

[123] Pismen, L.M. Nonlocal diffuse interface theory of thin films and the moving contact line. Physical Review E, Vol. 64, 021603.

[124] Premoze, S., Tasdizen, T., Bigler, J., Lefohn, A., Whitaker, R. Particle based simulation of fluids. Computer Graphics Forum (Proceedings of Eurographics 2003) 22 (3), pp. 401–410.

[125] Rasmussen, N., Nguyen, D.Q., Geiger, D.W., Fedkiw, R. Smoke simulation for large scale phenomena. SIGGRAPH 2003. ACM Transactions on graphics TOG, 22(3), pp. 703–707.

[126] Rhie, C.M., Chow, W.L. Numerical study of the turbulent flow past an airfoil with trailing edge separation, AIAA J., Vol.21 (1983), pp. 1525–1532.

[127] Roache, P.J. A flux-based modified method of characteristic. International Journal for Numerical Methods in Fluids. Vol. 15 (1992), pp. 1259–1275.

[128] Roberts, P.J.W., Webster, D.R. Turbulent Diffusion.

[129] Rosehead, L. The formation of vortices from a surface of discontinuity. Proc. Roy.Soc. London. Ser A, 134 (1932), pp. 170–192.

[130] Saffman, P. Vortex Dynamics. Cambridge University Press. London. 1992.

[131] Serrano, M., Espanol P. Thermodynamically consistent mesoscopic fluid particle model. Physical Review E,64(4), pp. 46115–,2001.

[132] Shepard, D. A two-dimensional interpolation function for irregularly spaced data. Proc. ACM Nat. Conf.1968.pp.517–523.

[133] Shu, C.W. Non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. ICASE Report No. 97-65. 1997.

[134] Siddiqi, K., Kimia, B.B., Shu, C. Geometric shock-capturing ENO schemes for subpixel interpolation, computation and curve evolution. Graphical Models and Image Processing. Vol. 59(5), September, pp.278–301, 1997

[135] Sims,K. Artificial evolution for computer graphics. Proceedings of SIGGRAPH'91, Computer Graphics, 25(4), pp. 319–328.

[136] Small, D. Modeling watercolor by simulating diffusion, pigment, and paper fibers. Proceedings of SPIE'91, February 1991.

[137] Smolarkiewicz, P.K., Pudikiewicz, J. A class of semi-Lagrangian approximations for fluids, J. Atmos. Sci. 49 (1992), pp.2082–.

[138] Stam,J. and Fiume.E. Turbulent wind fields for gaseous phenomena. Proceedings of SIGGRAPH '93. In Computer Graphics Proceedings, ACM SIGGRAPH, 1993, pp. 369–376.

[139] Stam J., Fiume E., Depicting fire and other gaseous phenomena using diffusion processes. ACM Computer Graphics, SIGGRAPH 95, pp.129–136, Addison Wesley, August 1995.

[140] Stam, J. Stable fluids. Proceedings of SIGGRAPH'99, in Computer Graphics, August, pp. 121–128.

[141] Stam,J. Flows on surfaces of arbitrary topology. Proceedings of ACM SIGGRAPH 2003 , pp. 724 – 731.

[142] Stein, M.L. Interpolation of spatial data: some theory for kriging, 1999.

[143] Steinhoff, J., Underhill, D. Modification of the Euler equations for "vorticity confinement": Application to the computation of interacting vortex rings. Physics of Fluids, Vol. 6, 1994, pp. 2738–2744.

[144] Stora, D., Agliati, P.O., Cani, M.P., Neyret, F., Gascuel, J.D. Animating lava flows. Graphics interface, 1999, pp. 203–210.

[145] Strassman, S. Hairy brushes. Proceedings of SIGGRAPH'86, Computer Graphics, 20(4), pp. 225–232.

[146] Sukumar, N., B. Moran and T. Belytschko (1998). The natural element method in solid mechanics, International Journal for Numerical Methods in Engineering 43, Number 5: 839–887.

[147] Sussman, M., Almgren A.S., Bell J.B., Colella P., Howell L.H., Welcome M.L. An adaptive level set approach for incompressible two-phase flows. Journal of Computational Physics. Volume 148 , Issue 1 (January 1999). Pages: 81 - 124.

[148] Sussman, F., Fatemi E. An Efficient, Interface-Preserving Level Set Redistancing Algorithm and Its Application to Interfacial Incompressible Fluid Flow. SIAM Journal on Scientific Computing. Volume 20 , Issue 4 (July 1999), pp. 1165 - 1191.

[149] Temam, R. Navier–Stokes equations: Theory and Numerical Analysis. North-Holland, Amsterdam, 1979.

[150] Terzopoulos, D., and Fleischer, K. Heating and melting deformable models (from goop to glop). Proceedings Graphics Interface, June 1989, pp. 219–226.

[151] Thevenaz, P., Blu, T., Unser, M. "Image interpolation and resampling." in Handbook of Medical Imaging, Processing and Analysis, I. Bancman, Ed. San Diego, CA, USA: Academic, 2000, ch. 25, pp.393–420.

[152] Tonnesen, D. Modeling liquids and solids using thermal particles. Proceedings of Graphics Interface'91, pp. 255–261.

[153] Unser, M. Splines: A perfect fit for signal and image processing. IEEE Signal Processing Magazine, vol. 16, no. 6, pp. 22–38, November 1999.

[154] Unser, M., Aldroubi, A., and Eden, M. B-Spline signal processing: Part I–Theory, IEEE Transactions on Signal Processing, vol. 41, no. 2, pp. 821–832, February 1993.

[155] Unser, M., Aldroubi, A. and Eden, M. B-Spline signal processing: Part II–Efficient design and applications. IEEE Transactions on Signal Processing, vol. 41, no. 2, pp. 834–848, February 1993.

[156] van der Waals, J.D. The thermodynamic theory of capillarity flow under the hypothesis of a continuous variation in density, Verhandelingen der Koninklijke Nederlandsche Akademie van Wetenschappen te Amsterdam, Vol.1 (1893). English translation: Rowlinson, J.S. Journal of Statistical Physics. 20. pp.197–244 (1979).

[157] VanDyke, M. An album of Fluid Motion. The Parabolic Press, Stanford, CA, 1983.

[158] Vorobieff, P., Rivera, M., Ecke, R.E. Soap film flows: Statistics of two-dimensional turbulence. Physics of fluids. Vol.11(8), August 1999.

[159] Watson, G.S. Smoothing and interpolation by kriging and with splines. Mathematical Geology, Vol. 16(6), pp. 601–615, 1984.

[160] Weiskopf, Dye Advection without the blur: A level-set approach for texture. SIGGRAPH 2004.

[161] Witkin, A., Kass, M. Reaction-diffusion textures. Proceedings of SIGGRAPH'91. In Computer Graphics (1991), vol.25, pp.299–308.

[162] Witting,P. Computational fluid dynamics in a traditional animation environment. SIGGRAPH 1999. pp.129–136.

[163] Wolfe, R.J. Marbled Paper: Its History, Techniques and Patterns. University of Pennsylvania Press. 1990.

[164] Woolnough, C.W. The Whole Art of Marbling. Oxford: Plough Press. 1985.

[165] Yabe, T., Xiao, F. Description of complex and sharp interface during shock wave interaction with liquid drop. J.Phys.Soc.Japan, vol. 62,no. 8,pp.2537–2540,(August 1993).

[166] Yaeger,L.,Upson,C.,and Myers,R. Combining physical and visual simulation- Creation of the planet Jupiter for the film "2010" ", SIGGRAPH'86, Computer Graphics 20(4),1986, pp. 85–93.

[167] Ye, Q.Z. The Signed Euclidean distance transform and its applications. Proc. 9th ICPR, Rome, Italy, 1988, pp. 495–499.

[168] Yepez, J. Lattice Gas Dynamics, Volume 1. Viscous Fluids Environmental research paper no. 1200. November 1995.

[169] Zhou, Y., McComb, WD, Vahala, G. Renormalization group (RG) in turbulence: historical and comparative perspective. ICASE Report No. 97-36, August 1997.